UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Rick Lopes de Souza

UM MIDDLEWARE PARA COMPARTILHAMENTO DE DOCUMENTOS SIGILOSOS EM NUVENS COMPUTACIONAIS

Florianópolis(SC)

2014

Rick Lopes de Souza

UM MIDDLEWARE PARA COMPARTILHAMENTO DE DOCUMENTOS SIGILOSOS EM NUVENS COMPUTACIONAIS

Dissertação submetido ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.
Orientador: Lau Cheuk Lung, Dr.

Florianópolis(SC)

2014

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Souza, Rick Lopes de
UM MIDDLEWARE PARA COMPARTILHAMENTO DE DOCUMENTOS SIGILOSOS EM NUVENS COMPUTACIONAIS / Rick Lopes de Souza ; orientador, Lau Cheuk Lung - Florianópolis, SC, 2014.
122 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Confidencialidade. 3. Nuvem. 4. Compartilhamento. 5. Middleware. I. Lung, Lau Cheuk. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Titulo.

Rick Lopes de Souza

UM MIDDLEWARE PARA COMPARTILHAMENTO DE DOCUMENTOS SIGILOSOS EM NUVENS COMPUTACIONAIS

Este Dissertação foi julgado aprovado para a obtenção do Título de "Mestre em Ciência da Computação", e aprovado em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis(SC), 11 de Agosto 2014.

Ronaldo dos Santos Mello, Dr.
Coordenador do Curso

Lau Cheuk Lung, Dr.
Orientador

Banca Examinadora:

Lau Cheuk Lung, Dr.
Universidade Federal de Santa Catarina

Fábio Favarim, Dr.
Universidade Tecnológica Federal do Paraná

Ricardo Alexandre Reinaldo de Moraes, Dr.
Universidade Federal de Santa Catarina

Ricardo Felipe Custódio, Dr.
Universidade Federal de Santa Catarina

1 AGRADECIMENTOS

Durante todo o período do mestrado obtive ajuda de diversas pessoas que de alguma forma foram especiais em minha vida. Primeiramente eu gostaria de agradecer a minha família por todo o apoio nos momentos difíceis e de luta durante esse período. Sem meus pais (Gerson e Maria Estela) e minha namorada (Dayanne Louise), com certeza essa jornada teria sido muito mais complicada. Graças a vocês eu pude ter forças para ir até o final.

Tenho de agradecer também aos colegas do LabSEC que sempre foram parceiros nas festas, churrascos e momentos de discussão de ideias. Tenho certeza que essa empreitada foi mais divertida e tranquila com vocês. Agradeço especialmente aos meus colegas Lucas Martins, Felipe Werlang, Guilherme Welter e Hendri Nogueira por tudo que vocês fizeram.

Tenho que agradecer de forma muito especial também o meu colega Hylson Vescovi que ficou muitas tardes discutindo comigo as ideias do trabalho e ajudando nas revisões. Você foi uma pessoa que me ajudou bastante e pode ter certeza que esse trabalho também é seu.

E por fim, gostaria de agradecer os professores Lau e Ricardo Custódio por terem me apoiado durante essa minha passagem pela Universidade Federal de Santa Catarina. Os dois professores me deram chance de mostrar o que eu era capaz e confiaram no meu trabalho.

Aprender é a única coisa de que a mente nunca se cansa, nunca tem medo e nunca se arrepende.

Leonardo da Vinci

RESUMO

Diversas empresas tem investido no uso da tecnologia da computação em nuvem para o armazenamento de documentos digitais e aplicações com o intuito de diminuir despesas com manutenção e infraestruturas físicas. Entretanto, o uso da computação em nuvem traz consigo vulnerabilidades que podem comprometer a sua ampla adoção no mercado. Documentos sigilosos normalmente são armazenados sem uma real preocupação com a segurança e, portanto, correm o risco de serem divulgados por indivíduos que tenham acesso físico ou administrativo no provedor de nuvem. O ciframento dos dados por meio da criptografia reduz o risco de determinadas ações, no entanto, aumenta a complexidade do sistema. Uma parte fundamental de tal sistema é o gerenciamento das chaves criptográficas. Mesmo havendo um sistema complexo para cifragem dos dados, caso o gerenciamento das chaves criptográficas seja falho, um indivíduo conseguirá realizar atos que podem comprometer a integridade e confidencialidade desses documentos. No entanto, desenvolver esse tipo de aplicação envolvendo computação em nuvem e gerenciamento de chaves criptográficas requer a integração e consideração de múltiplos aspectos de segurança e interoperabilidade. No caso de provedores de nuvem pública para armazenamento, a interoperabilidade entre diferentes provedores de nuvens para o armazenamento e controle de acesso é um desafio e deve ser transparente para o usuário. Isso também se aplica no gerenciamento de chaves criptográficas, um processo complexo e que necessita de segurança para garantir a confidencialidade de documentos. Para contornar esses desafios, esse trabalho tem como principal objetivo propor uma arquitetura de middleware para garantir o compartilhamento seguro de documentos sigilosos em nuvem utilizando provedores de nuvens públicas para o armazenamento dos dados e nuvens híbridas para o gerenciamento de chaves criptográficas. Este trabalho tem como principais características: o uso da criptografia baseada em identidade, uso de nuvens híbridas, gerenciamento de chaves criptográficas simplificado, garantia de segurança ponto a ponto e utilização de módulos de segurança criptográficos.

Palavras-chave: Sigilo. Nuvem. Compartilhamento. Armazenamento. Middleware.

ABSTRACT

Several companies have invested in the use of cloud computing technology for the storage of digital documents and applications with the aim of reducing costs maintaining physical infrastructure. However, the use of cloud computing brings with it vulnerabilities that can compromise its widespread adoption in the market. Classified documents are normally stored without a real concern for security, and therefore have the risk of having this information disclosed by individuals who have physical or administrative access to the cloud provider. The act of hide data through encryption reduces the risk of certain actions, however, increases the complexity of the system. A key part of such a system is the management of cryptographic keys. Even with a complex system for data encryption, if the cryptographic keys management is flawed, an individual can accomplish acts that may compromise the integrity and confidentiality of certain documents. However, developing such applications involving cloud computing and cryptographic key management requires the integration and consideration of multiple aspects of security and interoperability. In the case of public cloud storage providers, interoperability between different cloud providers for storage and access control is a challenge and should be transparent to the user. As in the management of cryptographic keys, a complex process that requires security to ensure the confidentiality of documents, which should keep these processes process transparent to users. To circumvent these challenges, this paper aims to propose a middleware architecture to ensure secure sharing of private documents using public cloud providers for data storage and hybrid clouds for managing cryptographic keys. This work has as main features: the use of identity-based encryption, use of hybrid clouds, simplified management of cryptographic keys, security assurance and peer use of cryptographic security modules.

Keywords: Secrecy. Cloud. Sharing. Storage. Middleware.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

2 INTRODUÇÃO

As vantagens em se utilizar os serviços de computação em nuvem trazem consigo o problema da garantia da privacidade dos dados armazenados. Usualmente, os servidores de computação em nuvem alocam mais de uma aplicação na mesma estrutura e, portanto, problemas como ataques advindos de outras corporações ou até mesmo de membros internos dos provedores de nuvem tornam-se possíveis de serem realizadas explorando possíveis vulnerabilidades dos sistemas envolvidos. Uma das principais preocupações das empresas em adotar os serviços de computação em nuvem é a garantia de segurança e privacidade de seus dados sensíveis (MESSMER, 2009). Não é difícil encontrar casos de dados roubados, como o caso da SalesForce em 2007, quando criminosos foram bem sucedidos em roubar informações de consumidores, como e-mails e endereços (GREENBERG, 2008). Para que usuários e corporações possam confiar nos serviços de armazenamento de dados sensíveis em nuvem, mecanismos que garatam as propriedades de confidencialidade, integridade, disponibilidade e autenticidade são necessários.

Para que seja possível restringir o acesso a dados sensíveis armazenados na nuvem, um controle de acesso seguro é necessário e as políticas de permissão devem restringir o acesso aos dados apenas àqueles que são autorizados pelos seus responsáveis. Se o sistema estiver alocado em apenas um provedor de nuvem, os donos dos dados sensíveis devem assumir que estes provedores são confiáveis e que irão prevenir o acesso contra usuários não autorizados. Para prover a segurança necessária sem depender tanto da confiança em terceiros, o provedor da nuvem não deve ter acesso aos dados em claro.

O uso da criptografia garante mais segurança no armazenamento de dados. Entretanto, se não houver um bom gerenciamento das chaves criptográficas, o sigilo das informações pode ser comprometido. Existem diferentes protocolos e arquiteturas para o gerenciamento dos documentos sigilosos em nuvem. Contudo, observa-se uma escassez nas soluções que abordam em conjunto as temáticas importantes para o compartilhamento de documentos sigilosos. Entre as principais necessidades pode-se listar a granularidade do uso das chaves, custódia das chaves por grupos, tolerância a falhas e gerenciamento seguro das chaves criptográficas por meio de módulos de segurança criptográficos.

Muitas técnicas foram propostas para tentar solucionar esses problemas, entretanto, não existe uma solução única para todos os casos. O esquema tradicional de uma infraestrutura de chaves públicas não é eficiente para resolver todos os problemas envolvidos no compartilhamento de informações

sensíveis e torna-se inviável quando o sistema cresce em números de usuários e armazenamento de dados devido a complexidade envolvida com carimbos de tempo (ELLISON; SCHNEIER, 2000). Outro tipo de solução para estes problemas é a técnica de Criptografia Baseada em Identidade (CBI) e foi primeiro introduzido por Shamir em 1985 (SHAMIR, 1985) e depois por Boneh e Fraklin com o protocolo *Identity Based Encryption* (BF-IBE) (BONEH; FRANKLIN, 2001). A CBI consiste em três entidades: emissor, receptor e uma terceira parte confiável. O emissor de uma mensagem especifica uma identidade (um conjunto de caracteres) de tal forma que apenas um receptor que corresponda a tal identidade consiga decifrar e ter acesso aos dados em claro. A autoridade confiável é responsável pelo processo de autenticação e por fornecer as chaves privadas necessárias. As chaves privadas estão diretamente ligadas às identidades dos usuários do sistema.

As autoridades confiáveis da CBI podem ter acesso às chaves privadas dos usuários e por esta razão, a técnica de CBI encontra resistências para ser implantada em alguns sistemas. Para superar esta limitação, múltiplas autoridades confiáveis devem ser utilizadas, de tal forma que nenhum destes tenham o poder total das chaves privadas dos usuários. O trabalho de Kate, Huang e Goldberg (2012) propõe um sistema de multi autoridades, no qual um conjunto de autoridades executa um algoritmo modificado do Joint Feldman Distributed Key Generator (JF-DKG) (GENNARO et al., 1999) para gerar a chave mestra secreta de uma forma distribuída. Usuários podem contatar um subgrupo de autoridades para solicitar uma parte da chave privada e então, depois de recuperar um número específico de partes, consegue reconstruir a chave privada em sua totalidade.

O desenvolvimento de aplicações que garantam o compartilhamento seguro de documentos sigilosos armazenados em nuvem é complexo e necessita integrar diversos protocolos de segurança. Divide-se em duas frentes os desafios encontrados. Uma das frentes encontradas é a parte de criptografia que necessita de protocolos e formatos específicos para a interoperabilidade de sistemas. Outra frente é a gerência de chaves criptográficas, um dos pontos críticos que pode comprometer a segurança de todo o sistema se não for projetada de forma correta. A outra frente é a parte de armazenamento em nuvens. Esse processo de armazenamento exige um estudo prévio de cada tecnologia e linguagem utilizada para que o armazenamento do sistema seja realizado de forma correta conforme os padrões de cada provedor. Além disso, existe também a questão da garantia de tolerância a falha e confidencialidade dos dados armazenados. Os provedores de nuvens públicas em geral fornecem poucos mecanismos para a garantia da confidencialidade e necessitam de conhecimento prévio para operacionalizar todos os protocolos criptográficos existentes.

Existem diversos trabalhos que tentam de maneiras diferentes e independentes garantir a confidencialidade no compartilhamento de documentos sigilosos em nuvem utilizando criptografia. Entretanto, as propostas existentes não fornecem os níveis de integração e segurança necessários no gerenciamento das chaves criptográficas em conjunto com o armazenamento eficiente e seguro de documentos. Dessa forma, existe a necessidade de uma proposta de um middleware que torne todo o processo de gerenciamento das chaves criptográficas seguro e transparente para usuários, assim como as operações criptográficas e armazenamento em nuvem. Dessa forma, pode-se fornecer uma interface de comandos simples para interagir com todos esses processos e garante-se a segurança no procedimento de compartilhamento de dados sigilosos em nuvem.

Este trabalho propõe uma arquitetura de middleware para compartilhamento seguro de documentos sensíveis em nuvem. A principal contribuição desta proposta é uma arquitetura que tornará o processo de compartilhamento seguro de documentos em nuvem transparente para as aplicações que o usuário utiliza, garantindo interoperabilidade entre diferentes provedores de nuvem de armazenamento e garantindo a segurança no gerenciamento de chaves criptográficas. Este trabalho possui as seguintes contribuições secundárias:

- Revogação segura de usuários;
- Gerenciamento seguro de chaves criptográficas utilizando módulos de segurança criptográficos;
- Tolerância a falhas;
- Emissão de chaves assimétricas por demanda;
- Reutilização do controle de acesso da aplicação que utilizar o middleware.

2.1 PERGUNTA DE PESQUISA

Este trabalho de pesquisa busca responder a seguinte pergunta: **Como garantir a integração de serviços de gerenciamento de chaves criptográficas e compartilhamento de documentos utilizando provedores de nuvens públicas de forma simples e segura?**

Para responder essa pergunta, sugere-se o uso de um middleware que utilize criptografia baseada em identidade em conjunto com múltiplos geradores de chaves privadas para o gerenciamento seguro de chaves criptográficas, assim como a distribuição de partes dos arquivos sigilosos em múltiplos

provedores de armazenamento em nuvem pública. Dessa forma, garante-se determinadas propriedades de segurança que viabilizam o compartilhamento de documentos sigilosos utilizando nuvens públicas. Para a garantia de um nível maior de segurança, sugere-se também o uso de módulos de segurança criptográficos, cuja função é proteger tanto fisicamente quanto logicamente as partes críticas do gerenciamento de chaves criptográficas. Ao utilizar as tecnologias citadas anteriormente, garante-se as seguintes propriedades: revogação segura de usuários, custódia segura das chaves criptográficas, tolerância a falhas e verificação de integridade confiável.

2.2 OBJETIVOS

Descreve-se nesta seção os objetivos gerais e específicos desse trabalho.

2.2.1 Objetivo Geral

Propor e avaliar uma arquitetura de middleware que proporcione o compartilhamento de documentos sigilosos de forma simples e segura.

2.2.2 Objetivos Específicos

- Elaborar mecanismos para gerenciamento seguro de chaves criptográficas utilizando criptografia baseada em identidade;
- Propor protocolos de integração de serviços de criptografia, gerenciamento de chaves criptográficas e armazenamento em provedores de nuvem públicas;
- Propor mecanismos para o armazenamento seguro e eficiente de documentos sigilosos em nuvem públicas;
- Propor uma arquitetura de middleware com uma API simples para o compartilhamento de documentos sigilosos em nuvem.

2.3 JUSTIFICATIVA

A computação em nuvem tem como principal obstáculo a garantia da segurança dos dados armazenados nos provedores públicos de nuvem para

armazenamento. Algumas corporações já utilizam esses meios para o armazenamento e para prover serviços. Entretanto, o problema geográfico do uso da computação em nuvem é algo recorrente. Como grande parte destes servidores estão localizados em outros países, estes estão sob outras jurisdições. Com isso, esses provedores de serviço de nuvem podem fornecer a terceiros dados sigilosos para outros interesses.

O uso da tecnologia da computação em nuvem vem crescendo nos últimos anos, contudo, a falta da garantia de segurança faz com que exista uma barreira para uma ampla adoção desta tecnologia. Logo, faz-se necessário a criação de novos mecanismos de segurança para que o uso dos provedores de nuvens públicas torne-se mais confiável.

O sigilo de documentos eletrônicos é de grande interesse às organizações devido a informatização de dados sigilosos. Entidades como tribunais de justiça, hospitais, órgãos governamentais e grandes empresas utilizam processos eletrônicos para o gerenciamento de seus documentos sigilosos, entretanto, grande parte dos esquemas utilizados não possui um forte esquema de segurança devido à escassez de soluções completas e com boa usabilidade para a preservação do sigilo em nuvem.

A criação de um middleware que diminua a complexidade e que aumente a segurança no desenvolvimento de aplicações que necessitem de sigilo ao utilizar provedores de nuvem pública é de interesse de diversas entidades. Existe a necessidade desse tipo de serviço devido a falta de propostas semelhantes e o crescente aumento do uso da tecnologia de computação em nuvem em conjunto com a falta de segurança no gerenciamento de documentos eletrônicos.

Existem sistemas descritos na literatura que viabilizam o acesso seguro de documentos na nuvem, mas comumente não abordam as políticas de acesso e a granularidade necessária das chaves criptográficas com detalhes. Observa-se também escassez nas soluções que garantem a segurança ponto a ponto e a custódia exclusiva de documentos sigilosos para grupos. Este último tópico é um grande desafio, pois em determinados sistemas um usuário final deve conseguir cifrar um documento para um grupo sem ter conhecimento exato de quem pertence ou vai pertencer a esse grupo.

Os tópicos acima exemplificam os fatores que tornam as soluções existentes complexas e difíceis de serem integradas a outros sistema. Por isso, este trabalho tem relevante importância devido as características de segurança e usabilidade que são garantidas e proporcionadas a outras entidades que desejam tornar seguro e menos complexo o compartilhamento de documentos sigilosos.

2.4 MOTIVAÇÃO

O estudo do presente trabalho se insere nas linhas de pesquisa do Laboratório de Segurança em Computação (LabSEC) e do Laboratório de Pesquisas em Sistemas Distribuídos (LaPeSD), locais do qual este trabalho foi desenvolvido. Esses laboratórios possuem projetos que envolvem sigilo de documentos eletrônicos, sistemas distribuídos e segurança em computação em nuvem.

Os problemas e desafios de segurança encontrados na utilização da computação em nuvem para armazenamento inibem o uso ainda mais amplo desta tecnologia. Desta forma, faz-se necessário o estudo de integração de mecanismos para prover a segurança na manipulação de arquivos sigilosos.

Após um levantamento na literatura especializada, percebeu-se que seria um trabalho oportuno de ser realizado para em uma dissertação de mestrado, fazendo com que fosse abordado temas relacionados a outros trabalhos do LabSEC trazendo também propostas de protocolos e mecanismos para que sistemas possam se adequar aos requisitos especificados.

2.5 LIMITAÇÕES DO TRABALHO

Existem diversos aspectos na segurança da computação em nuvem que podem ser considerados. Este trabalho limita-se a tratar apenas os desafios no gerenciamento das chaves criptográficas que garantem a confidencialidade no armazenamento de documentos. Os chamados “dados em descanso”, dados que estão armazenados, compõem o foco do trabalho.

Este trabalho limita-se a não entrar em tópicos como obsolescência de mídias ou arquivos e segurança em sistemas operacionais.

Este trabalho não implementa de forma integral todos os tópicos abordados no trabalho. Existe uma implementação como prova de conceito dos principais algoritmos de criptografia baseada em identidade para verificar sua aplicação e desempenho.

2.6 METODOLOGIA

Depois de estabelecida a problemática para a realização desde trabalho, inicialmente pesquisou-se na literatura, através de base de dados conhecidas como Google Scholar, ACM, IEEE, SpringerLinks e Mendeley, trabalhos relacionados, técnicas conhecidas e fez-se um levantamento dos trabalhos existentes sobre sigilo, armazenamento seguro em nuvem, gerenciamento de

chaves criptográficas e compartilhamento de documentos sigilosos.

Houve também uma análise das condições desses trabalhos relacionados, avaliando-se as soluções propostas e como se encaixavam em relação à problemática apresentada. Constatou-se uma escassez de trabalhos na literatura que abordassem os diversos problemas que estão intrínsecos no armazenamento seguro de documentos sigilosos na nuvem utilizando criptografia, como o compartilhamento de documentos e gerenciamento seguro das chaves criptográficas.

Por meio da compreensão dos mecanismos estudados e fazendo a relação entre as necessidades especificadas com a pesquisa dos trabalhos relacionados na literatura, protocolos e mecanismos foram elaborados para o gerenciamento seguro das chaves criptográficas para a garantia do sigilo de documentos eletrônicos armazenados em nuvem.

Depois de levantar os requisitos, premissas e objetivos, este trabalho propõe uma arquitetura de middleware para uso de aplicações que desejam diminuir a complexidade e aumentar a segurança no compartilhamento de documentos sigilosos em nuvem. Os protocolos propostos foram implementados e testados para uma avaliação de segurança e de desempenho, discutindo assim os benefícios e limitações apresentados.

2.7 ORGANIZAÇÃO DO TRABALHO

O Capítulo 2 deste trabalho apresenta os conceitos básicos que serão utilizados no decorrer dos trabalhos relacionados e propostas providas por esta dissertação. O Capítulo 3 contém descrições a respeito de trabalhos relacionados, comentando suas principais características e limitações. O Capítulo 4 contém a proposta em si, no qual os protocolos são detalhados, assim como os requisitos e premissas utilizadas. O Capítulo 5 contém a avaliação de segurança e desempenho da proposta apresentada no Capítulo 4. O Capítulo 6 contém as conclusões a respeito do trabalho e objetivos alcançados com a presente proposta, assim como os possíveis trabalhos futuros que podem agregar mais valor à proposta.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 INTRODUÇÃO

Faz-se necessário um capítulo para explicações sobre os conceitos básicos abordados neste trabalho, que tem o sigilo em nuvem como tema principal, antes de iniciar o tópico de trabalhos relacionados.

Este capítulo apresentará definições sobre segurança da informação, que são utilizadas ao longo do trabalho e também características críticas sobre informação. A seguir, o tema sigilo é tratado com mais detalhes com exemplificações de técnicas para garantir a sua existência. A problemática em relação ao gerenciamento de chaves é abordado, fazendo um levantamento crítico sobre os desafios na elaboração de protocolos. O armazenamento seguro de documentos eletrônicos é abordado, demonstrando sua importância no sigilo de documentos confidenciais.

3.2 SEGURANÇA DA INFORMAÇÃO

Documentos elaborados em papel podem adquirir a propriedade de sigilo e com isso, deve-se adotar técnicas para resguardá-los daqueles que não possuem a permissão de visualizá-lo. Tendo o documento a característica de ser algo concreto, como o papel, pode-se empregar técnicas como a esteganografia, que tem como função inserir uma mensagem secreta em outras mensagens (WILLIAM; STALLINGS, 2006). Porém, essa técnica não é suficiente.

Com o crescente processo de informatização de documentos, novas técnicas devem ser empregadas para garantir a segurança de dados sigilosos. Essa crescente procura pela informatização ocorreu devido às facilidades que a tecnologia proporciona, trazendo consigo, sustentabilidade, rapidez e segurança nos processos, gerando economias para grandes empresas e acelerando procedimentos que antes eram demorados. Por exemplo, processos físicos do Superior Tribunal de Justiça (STJ) do Brasil demoravam em média cem dias para serem distribuídos, e hoje, com processos eletrônicos, levam seis dias para chegar ao gabinete do relator (ITI, 2011).

O valor das informações varia conforme a característica do documento. Tudo depende do contexto em que é empregado e utilizado. Mas no ramo de segurança da informação, no qual existem dados confidenciais de extrema importância para determinadas entidades, órgãos ou nações, o valor das infor-

mações torna-se algo muito relevante. Segundo Whitman e Mattord (2010), existem algumas definições de características críticas quanto a informação:

- Disponibilidade: É a capacidade de fornecer ao usuário o acesso a informações sem nenhuma intervenção ou obstrução e também de recebê-las quando for requisitado.

- Precisão: Uma mensagem precisa é aquela que contém os dados livres de erros ou modificações e possui o sentido e valor que o usuário final espera.

- Autenticidade: Trata-se da qualidade ou estado da informação com relação à sua originalidade e se é genuíno. Um determinado documento pode se considerar autêntico caso esteja na mesma forma desde que foi criado, armazenado ou transmitido.

- Confidencialidade: Refere-se a documentos que exigem restrição quanto a sua divulgação ou exposição a pessoas não autorizadas. A propriedade de confidencialidade possibilita garantir que apenas determinadas pessoas tenham acesso ao conteúdo que é sigiloso. Existem diversas formas de se providenciar a confidencialidade, entre elas classificação da informação, armazenamento seguro de documentos, aplicação de políticas de segurança e treinamentos para os custodiantes de informações e usuários finais.

- Integridade: Uma informação possui integridade se está completa, inalterada e não corrompida. A integridade de uma informação estará comprometida se for exposta a corrupções, danos, destruições e modificações não autorizadas do seu estado autêntico. As alterações podem ser realizadas enquanto a informação é armazenada ou transmitida.

Para garantir determinados níveis de segurança, existem contrapartidas a serem realizadas, pois sempre que há um ganho na segurança, perde-se na liberdade, conveniência e tempo. Ao se desenvolver novos protocolos e mecanismos para garantir a segurança da informação de dados, deve-se averiguar quanto ao equilíbrio, observando a viabilidade de trocar a facilidade de certos processos por mecanismos mais complexos, e não quanto à eficiência da segurança que será empregada (SCHNEIER, 2008).

O conceito de segurança da informação possui uma ligação direta com fatores financeiros, tanto para marketing quanto produtos, no qual empresas vendem seus produtos informando sobre a qualidade do produto e passando a sensação de mais segurança, quanto para obter informações confidenciais e realizar futuras vendas para outras entidades. Isso normalmente é devido

ao gerenciamento pobre dos riscos, resultando em problema de privacidade e longas disputas regulatórias (ANDERSON; MOORE, 2006). Um exemplo sobre a venda de informações confidenciais é em relação a dados médicos, em que dados são comercializados com diretores de hospitais e companhias de seguro para traçar perfis e gerar novas campanhas de marketing alcançando um público alvo mais direcionado.

3.3 CONFIDENCIALIDADE

A confidencialidade é a propriedade de que um dado não é visível a determinadas entidades, a menos que tenham a autorização necessária para tal (SHIREY, 2007). Fazendo parte de um dos objetivos da criptografia, a confidencialidade é sinônimo de sigilo e privacidade e há diversas abordagens que podem garanti-la, como proteções físicas. Por exemplo, dispositivos criptográficos, controle de acesso e problemas matemáticos que tornam os dados ininteligíveis.

Estudos mostram que a criptografia não é uma área tão recente, pois há mais de quatro mil anos egípcios já utilizavam uma técnica de substituições de hieróglifos para deixar as mensagens com mais autoridade e autenticidade (KHAN, 1967). É por meio da criptografia, no qual utilizam-se algoritmos matemáticos para cifrar as mensagens, que se pode garantir a confidencialidade de dados. O procedimento de cifragem por sistemas de criptografia pode ser descrito através dos itens a seguir:

- Uma mensagem M em claro. Esta mensagem pode ser formada por caracteres, números, arquivos ou qualquer tipo de dado que se queira deixar em segredo;
- Uma mensagem C que representa a mensagem M cifrada. e não terá nenhum tipo de significado que possa ser relacionada com a mensagem original;
- Um conjunto de possíveis chaves κ;
- O procedimento de cifragem, que se caracteriza por transformar a mensagem original em uma mensagem cifrada, pode ser definida como: Ek: M $\rightarrow$ C, em que k $\in \kappa$;
- O procedimento de decifragem, responsável por recuperar a mensagem original, aplicando uma função inversa sobre o texto cifrado: Dk: C $\rightarrow$ M, em que k $\in \kappa$.

Conforme a definição de Denning e Elizabeth (1982), existem três pontos a serem considerados e seguidos por sistemas de criptografia:

- As funções de cifragem e decifragem devem ser eficientes para todos os tipos de chaves;
- O sistema deve ser de fácil uso;
- A segurança do sistema deve depender apenas do segredo das chaves e não do segredo dos algoritmos de cifragem e decifragem.

Os princípios de Kerckhoffs (1883) definem seis propriedades para sistemas de segurança, no qual parte deles serviram de inspiração e ainda são utilizados na criptografia moderna:

1. O sistema deve ser substancialmente, se não matematicamente, indecifrável.
2. O sistema não necessita de segredo e pode ser roubado pelo inimigo sem causar nenhum dano.
3. Deve ser fácil comunicar e lembrar as chaves, sem necessitar de nenhum tipo de anotação. Deve também ser fácil mudar ou modificar a chave com diferentes participantes.
4. O sistema deve ser portátil e não deve exigir mais de uma pessoa.
5. Finalmente, dependendo das circunstâncias em que o sistema é aplicado, deve ser de fácil utilização, não requerendo o conhecimento de demasiadas regras.

Uma das principais características que pode-se observar nas propriedades de Kerckhoffs é a ação de se manter em sigilo apenas as chaves criptográficas, deixando os algoritmos matemáticos em domínio público, propriedade essa ainda utilizada por grande parte dos algoritmos que proveem confidencialidade na atualidade.

Uma das garantias de segurança em sistemas que dependem de criptografia está baseado em problemas matemáticos de difícil solução, deixando o problema computacionalmente inviável. Para isso, o seu custo para execução depende de uma quantidade de memória incrivelmente grande, acima de qualquer máquina que possui um limite finito (DIFFIE; HELLMAN, 1976). Outra garantia que deve ser oferecida é de que ao obter o texto cifrado C e o seu texto em claro M, deve ser inviável descobrir a transformação de decifração.

3.4 GERÊNCIA DE CHAVES CRIPTOGRÁFICAS

Para garantir a segurança no armazenamento de dados sigilosos utilizando computação em nuvem, empresas adotam a criptografia como principal mecanismo para tornar os dados ilegíveis para aqueles que não possuem as chaves criptográficas necessárias. Entretanto, apenas parte do problema foi solucionado. Ao utilizar chaves criptográficas para cifrar documentos, deve-se manter um gerenciamento correto das mesmas para manter a segurança do esquema e para que o compartilhamento de documentos sigilosos seja simples, mesmo utilizando a criptografia.

Segundo o documento de Barker et al. (2011), ao lidar com a gerência de chaves criptográficas para grupos de usuários, existem alguns conceitos que devem ser considerados para a manutenção da segurança, como:

- ***Forward Secrecy:*** É uma propriedade na gerência das chaves que garante o sigilo de um sistema mesmo que um usuário saia do grupo. Ou seja, mesmo pertencendo a um grupo e tendo acesso a todas as chaves que cifram os documentos em um determinado momento, um usuário revogado não terá mais acesso a novos documentos que forem criados e cifrados após a sua exclusão.

- ***Backward Secrecy:*** Garante o sigilo e a manutenção de chaves e documentos exclusivamente a quem tem acesso aos mesmos, excluindo a possibilidade de que um membro que entre depois possa ter acesso sem autorização. Ou seja, caso um usuário seja adicionado em um grupo, documentos que foram cifrados antes da sua entrada e que não devem por alguma razão ser entregues a este novo membro, não deverão ser expostos.

- **Custódia da Chave:** É uma condição em que as chaves necessárias para decifrar dados criptografados são mantidas em custódia, de modo que, em certas circunstâncias, um terceiro não autorizado pode ter acesso a essas chaves. Esses terceiros podem incluir os provedores de nuvem, que podem querer acesso às comunicações dos usuários, ou governos, que podem desejar ser capaz de visualizar o conteúdo de comunicações criptografadas.

- **Revogação:** É uma ação da qual se exclui um usuário de um grupo, removendo assim a autorização para a obtenção de novas chaves criptográficas e a visualização de documentos sigilosos pertencentes a um grupo. Esta operação deve ser feita da maneira mais otimizada possível, evitando ter que recifrar todos os documentos para a remoção dos direitos do usuário que foi excluído.

- **Granularidade:** As chaves criptográficas gerenciadas pelo sistema devem apresentar boa granularidade, pois caso uma dessas chaves seja comprometida, poucos dados serão afetados. Ou seja, é desejável que as consequências do comprometimento de uma chave criptográfica sejam mínimas a tal ponto que apenas poucos arquivos sejam expostos.

- **Tolerância a Falhas:** Caso o gerenciamento de chaves seja feito por meio de um serviço de distribuição, deve-se estabelecer mecanismos de tolerância a falha. Um único ponto de falha pode comprometer o fornecimento das chaves e causar prejuízo a processos que dependam da disposição e uso imediato de chaves criptográficas.

- **Armazenamento das Chaves:** As chaves criptográficas devem ser armazenadas de forma segura. Esse armazenamento pode ser feito em dispositivos criptográficos que ficam de posse dos usuários ou em módulos de segurança criptográficos. O armazenamento é um dos pontos críticos no gerenciamento das chaves criptográficas devido ao risco de comprometimento em procedimentos mal realizados e ao comprometimento de dispositivos. Deve-se estipular mecanismos de cópias de segurança e procedimentos seguros para o correto gerenciamento.

- **Controle de Acesso:** Um dos principais pontos críticos no gerenciamento de chaves e integração com outros sistemas é o controle de acesso das chaves criptográficas. Esse tipo de gerenciamento é mais comum quando utiliza-se módulos de segurança criptográficos, no qual normalmente existe a necessidade da criação de usuários individuais responsáveis por chaves. Esse procedimento pode aumentar a complexidade na integração com outros sistemas e aumentar o risco de má gerenciamento das chaves criptográficas, podendo comprometer a segurança dos sistemas. Outro ponto crítico são os casos em que esse controle de acesso está sendo realizado pela nuvem e diretamente com os usuários. Isso faz com que dados de usuários como identidade, senha e outras informações sejam expostas desnecessariamente. As informações de usuários não devem ser expostas para os provedores de nuvem.

Essas propriedades expostas devem ser atendidas pelos sistemas que desejam garantir a segurança no compartilhamento de dados sigilosos em nuvem. Entretanto, são grandes os desafios devido à transparência que deve-se manter para os usuários. Desta forma, necessita-se empregar técnicas que possam prover estas garantias e ao mesmo tempo ser simples a tal ponto que os usuários tenham uma boa experiência de usabilidade.

3.5 MÓDULO DE SEGURANÇA CRIPTOGRÁFICO

O gerenciamento das chaves criptográficas é uma parte essencial de qualquer sistema que baseia-se em criptografia. A medida que o valor dessas chaves aumenta, necessita-se de um rígido controle de seus ciclos de vida. Para o armazenamento de chaves criptográficas, é necessário um dispositivo dedicado para gerenciar todo o seu ciclo de vida. Dispositivos que tem esse objetivo são chamados Módulos de Segurança Criptográfico (MSC).

Para proteger os dados armazenados internamente, conforme descrito por Sutil (2011), o MSC deve possuir uma área monitorada e protegida por meio de recursos lógicos e físicos. O principal objetivo da proteção dos MSCs é evidenciar qualquer tentativa de ataque realizado, como tentativas de corte, perfuração, ou qualquer outro tipo de ataque físico que possa comprometer os dados sensíveis. Esse perímetro de proteção recebe o nome de fronteira criptográfica.

Um módulo de segurança criptográfico consiste em três principais módulos: Software, Hardware e Firmware. O software tem como principal finalidade o gerenciamento das chaves criptográficas, contendo processos de autenticação e algoritmos criptográficos. O firmware tem como principal finalidade armazenar e gerenciar os recursos do sistema operacional. Esse firmware contém todos softwares necessários para o módulo de segurança criptográfico, como o software de gerenciamento das chaves e softwares para controle dos sensores do hardware. O hardware tem como principal finalidade proteger o conteúdo interno por meio de sensores que funcionam intermitentemente. Esses recursos ficam dentro da chamada “fronteira criptográfica”.

Um computador de propósito geral possui diversas aplicações de diferentes naturezas rodando em uma mesma memória principal compartilhada. Essa memória não conta com proteções contra ataques físicos. Um MSC mantém seus dados sensíveis protegidos pela fronteira criptográfica, com proteções físicas e lógicas. Grande parte de seus dados são persistidos de forma criptografada para a proteção dos dados. Além disso, o MSC dispõe de poucos aplicativos internos, para manter a segurança com base na simplicidade do sistema.

Não existe um tipo de padronização na construção de módulos de segurança criptográfico. Isso se deve principalmente pelo segredo de indústria das empresas que desejam ser superiores aos seus concorrentes e também para evitar que supostos entes maliciosos obtenham conhecimento a respeito de todos os mecanismos de segurança utilizados. Entretanto, existem alguns padrões que tem como principal objetivo manter um mínimo de interoperabilidade entre diferentes implementações. A Figura 1 representa uma arquitetura básica que um MSC teoricamente deve seguir, conforme a norma FIPS

140-2 (FIPS, 2001).

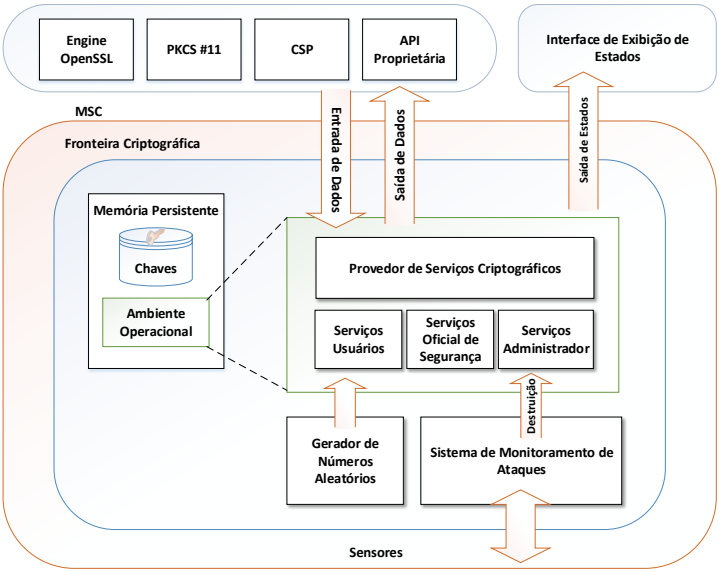

Figura 1 – Arquitetura genérica de um MSC.

Na a Figura 1 pode-se notar que todos os principais componentes do MSC estão dentro da fronteira criptográfica. Essa fronteira é composta por uma série de sensores, lacres e outros recursos que buscam proteger e evidenciar qualquer tipo de tentativa de ataque. Existe ainda um sistema de monitoramento interno e registro de intrusões que mediante a qualquer tentativa de intrusão, destruirá o conteúdo do MSC e quaisquer vestígios que possam comprometer a segurança do conteúdo armazenado. Em sua memória persistente, o MSC armazena as chaves criptográficas, dados de controle de acesso, assim como todos os outros aplicativos necessários para o seu funcionamento. O MSC, em geral, também deve dispor de um gerador de números aleatórios que é utilizado na geração de chaves criptográficas. Existe um provedor de serviços criptográficos que recebe e devolve as requisições dos clientes e dispõe de todas as funcionalidades de criação de chaves, execução de algoritmos criptográficos, assim como os processos de autenticação de usuários. Para realizar a comunicação entre clientes e o MSC, deve-se utilizar canais de comunicação confiáveis, utilizados para entrada e saída de dados, assim como a exibição de estados internos.

3.6 CRIPTOGRAFIA BASEADA EM IDENTIDADE

A criptografia baseada em identidade (CBI) foi proposta pelo trabalho (SHAMIR, 1985) com o objetivo de se utilizar conjuntos de caracteres, denominados identidades, como chaves públicas. Desta forma, não haveria a necessidade de se calcular aleatoriamente um número e vinculá-lo a uma pessoa por meio de um certificado digital. Ou seja, pode-se obter a chave pública de um indivíduo sem consultar nenhum repositório ou verificar certificados, basta ter o conhecimento do identificador relacionado. Um exemplo seria a utilização de um e-mail para a chave pública de um indivíduo.

Entretanto, esta proposta de Shamir concebia somente a assinatura baseada em identidade e não havia como utilizá-la nas aplicações reais para o ciframento baseado em identidade. Somente no ano de 2001 que o trabalho de Boneh e Franklin (2001) propuseram o primeiro esquema viável matematicamente de CBI, fazendo com que houvesse uma nova "corrida" na literatura por novas propostas de uso do CBI. A proposta de Bonneh baseia-se no uso do emparelhamento bilinear e possui as mesmas características das propostas por Shamir.

3.6.1 Emparelhamento Bilinear

Sejam dois grupos $G1$ e $G2$ de ordem q, sendo este um número primo grande. O emparelhamento bilinear se baseia na fórmula: $\hat{e} : G1 \times G1 \rightarrow G2$ entre os dois grupos. Para que o sistema de criptografia baseada em identidade seja viável, as seguintes propriedades devem ser providas:

1. Bilinear: Um mapeamento $\hat{e} : G1 \times G1 \rightarrow G2$ é dito bilinear se $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$ para todo $P, Q \in G1$ e para todo $a, b \in \mathbb{Z}$;

2. Não-Degenerativo: $\hat{e}(g,g) \neq 1$, no qual g é um gerador do grupo $G1$;

3. Computável: Existe um algoritmo eficiente para computar $\hat{e}(P,Q)$ para todo $P, Q \in G1$.

O mapeamento bilinear que satisfaz as três propriedades listadas acima é dito um mapeamento bilinear admissível. O grupo $G1$ é um subgrupo do grupo aditivo de pontos de uma curva elíptica $E/\mathbb{F}_p$. O grupo $G2$ é um subgrupo do grupo multiplicativo do corpo finito $\mathbb{F}^*_{p^2}$.

3.6.2 Conceitos Básicos

A criptografia baseada em identidade (CBI) baseia-se no uso de uma Autoridade de Confiança (AC) para que esta emita as chaves privadas referentes às identidades utilizadas como chaves públicas. Ou seja, um usuário que tem um identificador *ID* deve se comunicar com o sua AC e se autenticar. Caso essa autenticação seja feita de maneira correta, a AC envia por meio de um canal seguro a chave privada *CPriv* relacionada com o identificador *ID*. A CBI pode ser descrita com base nos seguintes passos, segundo Boneh:

- **Inicialização do Sistema:** Tem como entrada um parâmetro de segurança e retorna um conjunto de parâmetros públicos *PubParams* e uma chave mestra *S*. Entre os parâmetros públicos, deve-se conter a descrição do espaço finito que será utilizado para uma mensagem M e uma descrição do espaço finito de um texto cifrado C. Os parâmetros públicos serão publicamente divulgados enquanto a chave mestra será guardada e conhecida apenas pela AC. Explicação detalhada:

  - Passo 1: Ao receber um parâmetro de segurança $k \in \mathbb{Z}^+$, gera-se um primo q, dois grupos $G1$ e $G2$ de ordem q e um emparelhamento bilinear admissível $\hat{e} : G1 \times G1 \rightarrow G2$. Escolhe-se um gerador aleatório $P \in G1$.
  - Passo 2: Escolhe-se um número aleatório $s \in \mathbb{Z}_q^*$ e define-se como chave pública $P_{pub} = sP$.
  - Passo 3: Escolhe-se uma função de hash $H1 : \{0,1\}^* \rightarrow G1^*$. Escolhe-se outra função de hash $H2 : G2 \rightarrow 0,1^n$ dado um n. As análises de segurança veem $H1, H2$ como oráculos aleatórios. O espaço para as mensagens é $M = 0,1^n$. O espaço para mensagens cifradas é $C = G1^* \times \{0,1\}^n$. Os parâmetros públicos do sistema são $PubParams =$
    $< q, G1, G2, \hat{e}, n, P, P_{pub}, H1, H2 >$. A chave mestra $S \in \mathbb{Z}_q^*$.

- **Extração da Chave Privada:** Tem como entrada o conjunto de parâmetros públicos *PubParam*, a chave mestra S e um identificador arbitrário *ID* e retorna uma chave privada *CPriv*. O identificador *ID* é um conjunto de caracteres arbitrário que se utiliza como chave pública *CPub* e $Cpriv$ torna-se a chave privada correspondente. Explicação detalhada:

  - Dado um identificador $ID \in \{0,1\}^*$ o algoritmo realizará os seguintes passos:

  - Passo 1: Computa-se $Q_{ID} = H1(ID) \in G1^*$
  - Passo 2: Determina-se a chave privada $CPriv = sQ_{ID}$ no qual s é a chave mestra.

- **Cifragem:** Tem como entrada os parâmetros públicos *PubParam*, identificador ID e uma mensagem $Msg \in M$. O método retorna um texto cifrado $Cif \in C$. Explicação detalhada:
  - Para cifrar uma mensagem $Msg \in M$ com uma chave pública ID os seguintes passos são necessários:
  - Passo 1: Computa-se $Q_{ID} = H1(ID) \in G1^*$
  - Passo 2: Escolhe-se um número aleatório $r \in \mathbb{Z}_{11}^*$
  - Passo 3: O texto cifrado passa pelo seguinte procedimento: $Cif = < rP, M \oplus H2(g_{ID}^r) >$ no qual $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in G_2^*$.

- **Decifragem:** Recebe-se como entrada um texto cifrado Cif e uma chave privada $CPriv$. O método retorna o texto em claro $Msg \in M$. Explicação detalhada:
  - Seja $Cif = < U, V > \in C$ um texto cifrado utilizando a chave pública ID. Para decifrar Cif utilizando a chave privada $CPriv \in G1^*$ deve-se computar: $V \oplus H2(\hat{e}(CPriv, U)) = Msg$.

3.6.3 Inicialização do Sistema de forma Distribuída

No procedimento de inicialização do sistema nos protocolos de CBI, uma autoridade confiável irá gerar as chaves privadas para os usuários utilizando suas identidades públicas ID e uma chave mestra s. Um usuário com um identificador ID receberá a chave privada $CPriv = sH(ID)$, sendo H uma função de resumo criptográfico. Nesse tipo de inicialização do sistema, com o conhecimento da chave mestra s, a autoridade confiável possui o conhecimento e poder sobre as chaves privadas dos usuários e torna-se um único ponto de falha para o esquema de criptografia baseada em identidade. Para aperfeiçoar o modelo, utiliza-se um esquema de Geradores de Chaves Privadas distribuídos.

O mecanismo distribuído de geração de chaves baseado no protocolo de *Joint-Feldman Distributed Key Generator* (JF-DKG) (GENNARO et al., 1999) proposto por Kate, Huang e Goldberg (2012) atende as necessidades de uma geração distribuída e funciona de maneira simples e eficiente. O Protocolo JF-DKG requer um número $n \geq t$ de nodos para garantir uma geração de chave distribuída (n,t), no qual n é o número total de novos envolvidos e

t nodos honestos são suficientes para gerar corretamente o segredo, sendo o mais simples e eficiente dos geradores distribuídos de chaves.

O protocolo proposto por Aniket utiliza uma versão aprimorada do protocolo JF-DKG, chamado *Verifiable Secret Sharing* para gerar de forma distribuída a chave mestre. O protocolo propõe a inicialização do sistema por meio de um grupo de curvas elípticas G, com uma ordem q e um gerador U e o uso de um polinômio F(z) = $a_0 + a_1 z + ... + a_t z^t \in Z_q[z]$, tal que a chave mestra s = a_0. O protocolo de inicialização dispõe de um "quadro de avisos" que gera os parâmetros públicos para a inicialização do BF-IBE. Este será o polinômio compartilhado entre as partes, respeitando a premissa de que nenhum membro do grupo que participará do gerenciamento das chaves, tanto quanto o quadro de avisos conhece F(z) ou a chave mestra s.

Seja $s_j = F(j)$ o segredo compartilhado possuído pelo nodo P_j, tal que $j = 1,...,n$ e $Pub = sU$ é a chave pública correspondente. Um quadro de avisos online é criado contendo "pacotes de compromisso" verificados e assinados (pelos nodos do sistema) $A_k = a_k U$ para $k = 0,...,t$.

A partir dessas informações, pode-se seguir os seguintes procedimentos para a realização da inicialização do sistema:

- Dado um parâmetro de segurança k, o quadro de avisos escolhe um primo q de tamanho k, dois grupos G e G_T de ordem q e um emparelhamento bilinear $e : G \times G \to G_T$. O quadro de avisos escolhe uma função de resumo criptográfico H, escolhe um gerador aleatório $U \in G$ e torna esses dados públicos. Considerando que cada nodo possui seu polinômio $f_i(z) = a_{i0} + a_{i1} z + ... + a_{it} z^t$, são inicializados também os dados A_k e A_{ik} para zero, tal que $i = 1,...,n$ e $k = 0,...,t$. Consequentemente, a chave mestre s é configurada para zero.

- Nodos interessados em contribuir para a geração da chave mestra s devem inicializar o protocolo *Verifiable Secret Sharing* (VSS) modificado (KATE; HUANG; GOLDBERG, 2012). Como um adversário pode comprometer um máximo t nodos, uma vez que t nodos completam de maneira correta o protocolo, as partes distribuídas são consideradas seguras. Na literatura tais nodos são chamados de "nodos qualificados", ou simplesmente ∂.

- O quadro de avisos então computa e transmite os coeficientes A_k (para $k = 0,...,t$) para o polinômio compartilhado F(z).U, tal que $A_k = \sum_{P_j \in \partial} A_{ik}$.

- Após a verificação dos valores novos de A_k, os nodos enviam confirmações para o quadro de avisos.

- Ao receber t ou mais confirmações de assinatura, os valores A_k são finalizados. Cada nodo então computa sua parte do segredo $s_i = \sum_{P_j \in \partial} s_{ji}$.

Os nodos também obtém uma cópia assinada de A_k dos outros t ou mais nodos por meio do quadro de avisos.

3.6.4 Extração da Chave Privada

Após o processo bem sucedido de inicialização do sistema, os nodos geradores de chaves privadas estão preparados para extrair as partes das chaves privadas para os usuários. Assume-se que no mínimo t de um total de n nodos estarão disponíveis para os usuários. Seja ∂ o conjunto t de servidores disponíveis e escolhidos pelo cliente. O protocolo de extração das chaves privadas funciona da seguinte maneira:

- Um usuário com um identificador ID comunica-se com um nodo disponível do conjunto ∂.
- Cada nodo P_i verifica a identidade do usuário e retorna uma parte da chave privada $s_i H(ID)$ por meio de um canal seguro e autenticado.
- Após receber t partes corretas da chave privada, o usuário pode reconstruir sua chave privada d_{ID} utilizando $d_{id} = \sum_{P_i \in O} \lambda_i s_i H(ID)$, no qual o coeficiente de Lagrange é $\lambda_i = \prod_{P_j \in O, j \neq i} \frac{j}{j-i}$.
- O usuário pode verificar a corretude da sua chave privada computada d_{ID} utilizando $e(d_{ID}, U) = e(H(ID), Pub)$. Caso não se encontre a igualdade, o usuário poderá verificar a corretude de cada parte recebida $s_i H(ID)$ verificando se $e(s_i H(ID), U) = e(H(ID), Pub_i)$. Uma igualdade prova a corretude da parte, enquanto a diferença indica um mau comportamento de um nodo P_i.

3.7 COMPUTAÇÃO EM NUVEM

A computação em nuvem é um conceito recente sobre a utilização de memória, armazenamento e capacidades de computação feitos por computadores interligados de forma distribuída, seguindo o princípio da computação em grade. O principal objetivo da computação em nuvem é dar acessibilidade a seus provedores de conteúdo de qualquer lugar do mundo, utilizando os mais variados dispositivos, não havendo a necessidade de instalar componentes ou armazenar dados.

O uso da computação em nuvem consegue satisfazer financeiramente grande parte das empresas, sejam elas de pequeno, médio ou grande porte,

quando o propósito final é a disponibilidade dos dados e serviços. Essa tecnologia consegue diminuir os investimentos das empresas no quesito infraestrutura física, fazendo com que essas invistam apenas no que realmente estão utilizando, sem ter a necessidade de ocupar locais físicos e comprar hardwares específicos.

3.7.1 Características

De acordo com o documento de Mell e Grance (2011), são cinco as principais características da computação em nuvem:

- **Serviço sob demanda:** Um usuário pode utilizar recursos da nuvem, como tempo de uso ou espaço para armazenamento, como for necessário, de forma automática sem necessitar de interação humana com cada provedor de serviços.

- **Acesso de rede amplo:** Os recursos estão disponíveis por meio da Internet, acessados por mecanismos padrões que promovem o uso por diferentes plataformas, como *smartphones*, *tablets*, *workstations* e *notebooks*.

- **Disponibilidade de recursos:** Os recursos da computação em nuvem são agrupados para atender a vários consumidores por meio de um modelo *multi-tenant*, com diferentes recursos físicos e virtuais atribuídos dinamicamente de acordo com a demanda do consumidor.

- **Escalabilidade:** A capacidade do provedor da nuvem pode ser elasticamente adquirida e liberada e, em alguns casos, de forma automatizada para rapidamente utilizar mais ou menos recursos de acordo com a demanda. Para o consumidor, as capacidades disponíveis parecem ser ilimitadas e podem ser utilizadas em qualquer quantidade a qualquer momento.

- **Serviço de medição de recursos:** Os sistemas de computação em nuvem controlam de maneira automática o uso de recursos. Por exemplo, no caso da computação em nuvem pode-se utilizar: armazenamento, processamento, largura de banda e contas de usuários. O uso dos recursos pode ser monitorado, controlado e reportado, oferecendo transparência tanto ao provedor do serviço da nuvem quanto ao consumidor dos serviços utilizados.

3.7.2 Vantagens

A utilização dos serviços na nuvem pode trazer diversos benefícios aos fornecedores de conteúdo. Alguns destes benefícios são listados abaixo:

1. **Flexibilidade**: Ao necessitar de mais recursos, os provedores de nuvem podem automaticamente atender a esta demanda devido à vasta capacidade dos servidores remotos. Grande parte das empresas necessitam deste tipo de flexibilidade e por isso estão migrando para a computação em nuvem (WEEK, 2008).

2. **Recuperação de Desastres**: Ao migrar os dados e serviços para a nuvem, as grandes corporações não necessitam mais se preocupar com planos de recuperação de desastres complexos. Uma vez que os dados estão na nuvem, estes se encarregam de gerenciar a segurança dos dados, fazendo cópias de segurança e replicando entre outros servidores. Dessa forma, torna-se esse tipo de serviço mais confiável e viável financeiramente para as empresas que necessitam montar seus próprios planos de recuperação.

3. **Rápida Implantação**: A computação em nuvem possibilita uma rápida implantação. Ao contratar determinados tipos de serviços, o cliente poderá optar por aquele que mais se adeque as suas necessidades e poderá implantar a sua aplicação de um modo funcional em minutos.

4. **Gastos sob Demanda**: A computação em nuvem tem como uma de suas principais características a viabilidade econômica. Esta viabilidade é devido ao fato de que as empresas normalmente gastariam muito dinheiro para ter toda uma infraestrutura interna com as mesmas funcionalidades. Ao utilizar recursos de gastos sob demanda, as empresas acabam gastando apenas aquilo que estão utilizando, tendo a opção de contratar mais caso necessite.

5. **Aumenta a Colaboração**: A computação em nuvem aumenta a colaboração entre os funcionários de empresas, seja onde estiverem, para sincronizar seus trabalhos, documentos e dados de uma forma simultânea, fazendo com que os receptores consigam obter atualizações em tempo real. Um *survey* feito por Frost e Sullivan mostra que as empresas que investiram em tecnologia de colaboração tiveram 400% de retorno do investimento (CISCO, 2010).

6. **Mobilidade**: Um dos principais benefícios da utilização da computação em nuvem é a mobilidade que os usuários podem ter. Os usuários

que tiverem acesso a internet, podem acessar seus dados e trabalhar de qualquer lugar. Este tipo de mobilidade e flexibilidade afeta de maneira positiva a produtividade e a qualidade de vida de muitos usuários.

7. **Compartilhamento de Documentos**: De acordo com uma pesquisa da empresa Adobe, 73% dos trabalhadores hoje compartilham documentos de alguma forma com outras pessoas em diferentes regiões. Ao utilizar a computação em nuvem, este tipo de compartilhamento é facilitado pela mobilidade e flexibilidade que esta tecnologia proporciona (ADOBE, 2011).

3.7.3 Modelos de Serviços

Existem três principais modelos de computação em nuvem: Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS). Abaixo são listadas as principais características de cada um destes modelos de serviços.

- **Infraestrutura como Serviço (IaaS)**: O modelo de infraestrutura como serviço tem como principal característica a disponibilização de computadores físicos ou máquinas virtuais, sendo este último o mais comum. Este tipo de serviço normalmente também oferece bibliotecas para trabalhar com imagens de disco, armazenamento baseado em arquivo, segurança com *firewalls*, balanceadores de carga e esquemas de *virtual local area network* (VLAN). Exemplos de IaaS: Amazon EC2, Google Compute Engine e HP Cloud.

- **Plataforma como Serviço (PaaS)**: Este tipo de modelo fornece uma plataforma de computação e uma pilha de soluções. Neste tipo de modelo o cliente cria os softwares utilizando as ferramentas e bibliotecas disponibilizadas pelo provedor da nuvem. O cliente também controla a implantação do software e as configurações do sistema. O provedor fornece a infraestrutura de rede, servidores, armazenamento e outros serviços. Exemplos de Paas: Heroku, Google App Engine e Windows Azure.

- **Software como Serviço (SaaS)**: Neste modelo os usuários finais tem acesso aos serviços disponibilizados pela nuvem. Os provedores da nuvem gerenciam a infraestrutura e a plataforma que rodam os serviços finais. O modelo SaaS normalmente é referenciado como software sob demanda, no qual os usuários finais pagam por uso. Com este tipo de modelo, os usuários finais não tem a necessidade de instalar qualquer

tipo de software em suas máquinas locais, o que simplifica a manutenção e suporte. Os processos de atualização dos softwares e recuperação de segurança são transparentes aos usuários finais. Exemplos de SaaS: Google Apps e Salesforce.

3.7.4 Modelos de Implantação

Os modelos de implantação dependem das necessidades das aplicações que serão implementadas. O controle de acesso depende das regras de negócio, tipo de informação e interesse desejado. Algumas entidades podem não desejar que os seus recursos sejam acessados por todos os usuários de qualquer lugar. Os diferentes tipos de implantação, de acordo com o documento Mell e Grance (2011) estão listados a seguir:

- **Privado:** As nuvens privadas são infraestruturas operadas em um perímetro controlado. Normalmente esta restrição acontece dentro de uma única organização no qual pode ser gerenciada e operada pela mesma, uma terceira parte ou uma combinação entre elas.
- **Público:** As nuvens públicas são infraestruturas disponibilizadas para o público em geral. Pode ser de propriedade, gerenciada e operada por organizações privadas, acadêmicas, governamentais ou uma combinação entre elas. Ela existe nas instalações do provedor da nuvem.
- **Híbrido:** A infraestrutura híbrida é uma composição entre duas ou mais infraestruturas de nuvem (privada ou pública) mantendo-se como entidades únicas, no entanto, são unidas por tecnologias padronizadas ou proprietárias que permitem a portabilidade de dados e aplicativos.

3.7.5 Ameaças à Computação em Nuvem

Uma das grandes ameaças à computação em nuvem é o controle que os provedores deste tipo de tecnologia detém sobre as comunicações e dados dos usuários com as empresas contratantes. Recentemente os Estados Unidos, por meio da agência *NSA*, espionaram milhões de ligações, comunicações e dados armazenados nos mais variados serviços de nuvem. Ou seja, esse tipo de mecanismo dá muitos poderes às empresas de telecomunicações e provedores de nuvem que monitoram as atividades dos seus usuários. Exemplos de empresas que tem parceria com o governo americano e que fornecem dados para as espionagens: Google, Microsoft, Apple, Facebook, entre outras (GREENWALD; MACASKILL, 2013).

Localização dos dados: Um dos principais problemas relacionados à computação em nuvem é a geolocalização dos provedores de nuvem. Esses normalmente estão localizados em outros países, como os Estados Unidos, no qual possuem rígidos controles de dados para controles internos e fazem com que os dados das corporações de outras empresas fiquem expostos a interesses destes países (JAEGER et al., 2009). O usuário final normalmente não sabe onde estão armazenados seus dados e, com isso, terá dificuldades de buscar por seus direitos caso seus dados sejam expostos devido as diferentes leis em cada país.

Segurança da Rede: Na computação em nuvem os dados são obtidos das estações locais dos usuários, processados e armazenados na nuvem. Todos os dados que trafegam entre o usuário e a nuvem precisam ser cifrados com o objetivo de proteger os dados com conteúdos sensíveis. Para que isso seja feito, deve-se utilizar mecanismos de cifragem como o *Secure Socket Layer* (SSL). Estes provedores de serviço de nuvem devem estar protegidos contra ataques advindos da rede, como por exemplo, ataque do homem no meio, *IP spoofing* e captura de pacotes.

Segregação de Dados: Uma das principais características da computação em nuvem são os múltiplos clientes que acessam a mesma instância do software na nuvem. Como resultado deste uso compartilhado, dados de diversos usuários serão armazenados no mesmo servidor. Com isso, ataques de invasão ao espaço de outros usuários tornam-se possíveis pelo não-isolamento seguro dos dados. Estes tipos de ataques podem ser realizados inserindo códigos maliciosos no provedor de serviços. Se o provedor executá-los sem uma verificação, existe um alto potencial de sucesso no ataque. O provedor de serviços deve prover mecanismos de separação segura entre os dados dos usuários.

Autenticação e Autorização: Grande parte das empresas que utilizam a computação em nuvem armazenam as informações e credenciais de seus funcionários nesses provedores de serviços. Com isto, os dados ficam fora das proteções da empresa, tornando assim os dados dos funcionários vulneráveis a possíveis ataques. Uma das alternativas seria a utilização de parte destes serviços para os servidores internos da empresa. Desta forma, alguns dados sensíveis e autenticações poderiam estar protegidos física e logicamente.

Confidencialidade dos Dados: Quando indivíduos, empresas ou governos utilizam a computação em nuvem para armazenar seus dados, questões sobre a privacidade e confidencialidade são discutidas. Pela natureza distribuída da nuvem, os dados dessas organizações são compartilhadas em diversos servidores que são de propriedade externa e que são operados por terceiros. Existem diversos serviços de nuvem que armazenam dados sensíveis, como sistemas governamentais, de saúde, judiciários e sistemas pessoais. Al-

gumas das implicações relacionadas a privacidade são:

- A preocupação com o sigilo dos dados não se restringe a dados pessoais, mas também de empresas e órgãos governamentais;
- Os direitos sobre a privacidade e confidencialidade dos dados muda conforme os tipos e categorias de dados que o provedor de conteúdo fornece para a nuvem;
- O armazenamento de dados pessoais e empresariais feito em nuvens pode gerar consequências negativas para a imagem com relação à privacidade dos dados;
- A localização dos servidores tem influência direta sobre os efeitos da privacidade e obrigações legais para aqueles provedores que fornecem serviços e armazenam dados;
- As informações armazenadas na nuvem podem estar localizadas em mais de um local, podendo assim estar sob jurisdições diferentes;
- Algumas leis podem obrigar os provedores de nuvem a fornecer dados de seus usuários alegando investigações sobre crimes ou outros motivos;

Falhas: Servidores dos provedores de nuvem podem falhar por diversas razões e parar de oferecer seus serviços. Um sistema que preza pela segurança deve ser tolerante a falhas, mantendo o funcionamento mesmo que em menor capacidade. Essa propriedade é uma contínua ameaça aos provedores de nuvem, haja visto que o usuário não detém controle sobre o funcionamento total do serviço contratado. Por esse motivo, deve-se utilizar uma arquitetura diferenciada, que possa manter o funcionamento do sistema em casos de falha. Uma das alternativas que pode-se adotar é a utilização de múltiplos provedores de nuvem, garantindo assim uma maior heterogeneidade no fornecimento dos serviços e minimizando o impacto de possíveis falhas dos sistemas.

Uma das formas de se lidar com a questão da tolerância a falhas é o uso de mecanismos de arquivos compartilhados entre k servidores, de tal forma que é necessário um número mínimo n para recompor o arquivo original. Uma das técnicas que normalmente é utilizada para esse tipo de ameaça é o *Secret Sharing*. Esse mecanismo delimita um número mínimo de k partes entre um total de n para recompor um determinado segredo. Baseia-se no uso de $k-1$ coeficientes de um polinômio $F(x) = a_0 + a_1x + a_2x^2 + ... + a_{k-1}x^{k-1}$. Então n pontos são construídos a partir do polinômio, gerando as partes dos usuários. Esses conseguem reconstruir suas partes utilizando a interpolação

de Lagrange. Entretanto, ao utilizar essa técnica, cada parte gerada possui a mesma quantidade de bytes que o segredo a ser quebrado. Dessa forma, de um segredo com T bytes, gera-se um total de n partes cujo tamanho é de T bytes cada. Ou seja, o total de armazenamento utilizado seria um total de nT bytes.

Existe uma outra técnica que consegue diminuir essa quantidade de dados armazenados e apresenta um desempenho satisfatório. A técnica chama-se *Erasure Code* (RABIN, 1989) e baseia-se na transformação de uma determinada mensagem em uma mensagem maior de tal forma que a mensagem original pode ser reconstruída por um subconjunto de n partes. Essa técnica é mais utilizada para o controle de erros na comunicação de dados por meio de um canal não confiável e com ruídos. A ideia central é que o emissor codifica a mensagem de uma forma redundante utilizando um código de correção de erro. Ao determinar um número de partes mínimas k para recompor a mensagem, o método gera $n-k$ partes redundantes. Ao combinar quaisquer k partes, pode-se recompor a mensagem original. No entanto, cada parte possuirá por volta de metade do tamanho original da mensagem. Ou seja, uma mensagem cujo tamanho é de T bytes, será dividida em n partes de $T/2$ bytes cada. Finalmente, o espaço total de armazenamento que será utilizado com essa técnica seria de $n(T/2)$ bytes, apresentando assim uma economia no armazenamento de dados distribuído em diferentes provedores de nuvem.

3.8 CONCLUSÃO DO CAPÍTULO

Para garantir a segurança no compartilhamento de documentos sigilosos, existe a necessidade de prover e adotar as características e procedimentos listados anteriormente sobre o gerenciamento de chaves criptográficas. Necessita-se também observar os riscos apresentados ao utilizar a tecnologia da computação em nuvem e verificar quais são as medidas mais adequadas a serem utilizadas em determinadas situações específicas.

Ao lidar com a situação de compartilhamento de documentos sigilosos, necessita-se de um gerenciamento de chaves criptográficas seguro e que seja simples. O uso da criptografia baseada em identidade pode prover um esquema de gerenciamento seguro e ao mesmo tempo simples ao utilizar como chaves públicas identificadores específicos em forma de texto. Entretanto, ao utilizar a criptografia baseada em identidade esbarra-se no problema de custódia das chaves privadas. Consequentemente, deve-se utilizar um mecanismo de geração de chaves distribuídos, agregando assim mais segurança no esquema.

4 TRABALHOS RELACIONADOS

4.1 INTRODUÇÃO

Neste capítulo serão discutidos os principais trabalhos disponíveis na literatura que estão relacionados diretamente com o tema desta dissertação. Os temas serão divididos da seguinte forma: Sigilo como Serviço, Gerenciadores Locais de Sigilo e Sigilo com Criptografia Baseada em Identidade. Serão descritas as principais características, contribuições e limitações de tais trabalhos.

O primeiro grupo contém trabalhos que apresentam soluções para o sigilo na nuvem baseado em serviços. Estes trabalhos tem como principal característica um servidor que executará serviços que tornarão os dados sigilosos.

O segundo grupo apresenta trabalhos que propõem o sigilo realizado nas máquinas dos usuários. Este grupo tem como principal característica a ação local do usuário para tornar o arquivo sigiloso.

O terceiro grupo contém trabalhos de sigilo em grupo baseados no uso da criptografia baseada em identidade. Esses trabalhos utilizam o conceito da chave pública baseada em uma identidade, tentando assim facilitar o gerenciamento das chaves criptográficas. Por fim, tem-se as conclusões a respeito dos trabalhos apresentados.

4.2 SIGILO COMO SERVIÇO

Existem trabalhos que propõem a ideia do sigilo como serviço, no qual existe uma terceira parte que fica encarregada de cifrar ou decifrar arquivos conforme políticas de acesso. Nesta abordagem, parte da segurança ponto a ponto é garantida em forma de serviço. Os trabalhos abordam de maneiras diferentes esses temas, como um serviço integral de sigilo, em que o usuário envia o documento em claro para esta terceira parte que então cifra e armazena, e também os que funcionam de maneira conjunta, tendo participação tanto do usuário quanto da terceira parte confiável.

Um dos trabalhos que utilizam o sigilo como serviço de maneira conjunta foi publicado por Ateniese et al. (2006). Este utiliza uma terceira parte que funciona como um "proxy" que recifra documentos para que outros usuários possam decifrar e ter acesso aos conteúdos sigilosos.

A abordagem de "proxy" recifragem de documentos sigilosos (BLAZE;

BLEUMER; STRAUSS, 1998) consiste em delegar o poder de recifragem a uma terceira parte semi-confiável. Este procedimento consiste no fato de que esta entidade semi-confiável consegue transformar um texto cifrado de um usuário A em um texto cifrado para o usuário B sem ter acesso ao conteúdo em claro.

O trabalho de **??**)ateniese2006improved) utiliza este conceito de "proxy" recifragem para que um usuário dono de um arquivo compartilhe um documento sigiloso com outros usuários. O trabalho propõe um esquema que possui os seguintes métodos:

- Gerar Chaves: Retorna uma chave pública $pk_a = (Z^{a1}, g^{a2})$ e uma privada $sk_a = (a_1, a_2)$;
- Gerar Chaves de Reciframento: Um usuário A gera uma chave de reciframento para B publicando a chave $rk_{AB} = g^{a_1 b_2} \in G_1$ computado por meio das informações públicas de B;
- Ciframento de Primeiro Nível: Para que um usuário A cifre uma mensagem m de tal forma que apenas A tenha acesso, a saída será $c_{a,1} = (Z^{a_1 k}, mZ^k)$.
- Ciframento de Segundo Nível: Para que um usuário A cifre uma mensagem m de tal forma que A e os delegados tenham acesso, a saída será $c_{a,r} = (g^k, mZ^{a_1 k})$.
- Reciframento: Qualquer um que tenha acesso a chave de reciframento rk_{AB} pode transformar uma mensagem cifrada de segundo nível para A em uma de primeiro nível para B. Para isso, dado $c_{a,r} = (g^k, mZ^{a_1 k})$, computar $e(g^k, g^{a_1 b_2}) = Z^{b_2 a_1 k}$ e publicar $c_{b,2} = (Z^{b_2 a_1 k}, mZ^{a_1 k})$.
- Deciframento: Para decifrar uma mensagem cifrada $c_{a,i} = (K, L)$ com uma chave secreta $a_i \in sk_a$, computar $m = L/K^{1/a_i}$ tal que $i \in 1, 2$. Para decifrar uma mensagem cifrada de segundo nível $c_a = (K, L)$ com uma chave secreta $a_1 \in sk_a$, computar $m = L/e(K, g)^{a_1}$.

O trabalho propõe o uso de três tipos de entidades: Servidor de Armazenamento, Usuário e Servidor de Controle de Acesso. Por meio destas entidades, um usuário A pode compartilhar um documento sigiloso com B por meio do seguinte esquema:

1. Um usuário B publica sua chave pública pk_B;
2. Usuário A importa a chave pública pk_B de B e gera uma chave de reciframento rk_{AB}.

3. Esta chave de reciframento é então exportada para uma terceira parte semi-confiável que fará o controle de acesso e o 'proxy' reciframento.

4. O usuário *A* cifrará com o primeiro e segundo nível um determinado documento *DOC* e publicará em um servidor de armazenamento para que *B* consiga futuramente obter este documento e decifrá-lo.

5. O usuário *B* busca o documento cifrado do servidor de armazenamento e autentica-se perante o servidor de controle de acesso. Este último, ao verificar a identidade de *B*, recifra o documento para que apenas *B* consiga decifrá-lo por meio da sua chave privada.

Com estes métodos exemplificados, o trabalho de **??**)ateniese2006improved) consegue cumprir com as seguintes propriedades de segurança: o compartilhamento de documentos sigilosos entre usuários e grupos, a revogação segura de usuários por meio da renovação das chaves de recifração e o servidor não possui a custódia das chaves privadas, tornando-o assim incapaz de visualizar os conteúdos cifrados. Entretanto, este trabalho possui algumas limitações com relação a complexidade do compartilhamento dos documentos. Para que usuários compartilhem documentos sigilosos, estes devem compartilhar entre si as chaves públicas para a geração de novas chaves mestres. Ou seja, para cada novo usuário que for compartilhado um arquivo, será sempre necessário receber uma nova chave pública para gerar novos parâmetros de compartilhamento.

Devido a essa característica, o trabalho apresentará problemas para um compartilhamento com uma granularidade alta, como quando deseja-se compartilhar um documento sigiloso com um grupo de usuários sem ter o exato conhecimento de quem faz parte deste. Portanto, este modelo possui restrições nas mudanças de grupo e no compartilhamento dos documentos sigilosos.

O trabalho de Xiong et al. (2012) também utiliza o modelo de recifragem. O trabalho apresenta um modelo com três entidades: provedor de conteúdo, o provedor da nuvem e usuários receptores. O trabalho confia somente no provedor de conteúdo e nos usuários receptores. Dessa forma, o modelo não libera materiais sigilosos e chaves criptográficas de decifragem para terceiros não autorizados e possui mecanismos para revogação do usuário, removendo fisicamente as chaves dos usuários revogados. O provedor da nuvem é considerado semi-confiável, seguindo os protocolos definidos pelo trabalho, contudo, pode ativamente tentar descobrir o conteúdo dos dado sigilosos.

De maneira resumida, os objetivos e a segurança do sistema são os seguintes:

- Deve garantir a confidencialidade dos dados, mesmo no caso de conluio entre o provedor da nuvem e os usuários receptores;
- Deve suportar estados dinâmicos, em que usuários podem entrar e sair de grupos e serem revogados de um grupo pelo provedor de conteúdo;
- Deve garantir o sigilo contra membros que não estão no grupo e contra membros que saíram no grupo. Ou seja, um membro que entrar em um grupo, não poderá ter acesso aos documentos que faziam parte antes deste entrar. Assim como este usuário não poderá mais ter acesso aos dados sigilosos após a sua remoção ou revogação de um grupo.

Existem três entidades nesse modelo: Provedor da Nuvem, Provedor do Conteúdo e Usuários Receptores. As funções de cada um podem ser definidas como:

- Provedor da Nuvem: Provê dois tipos de serviços: um serviço de armazenamento e um serviço de distribuição de conteúdo. Também provê uma infraestrutura virtual para hospedar aplicações que podem ser utilizadas pelos provedores de conteúdo para manipular dados armazenados na nuvem e usuários receptores para recuperar dados.
- Provedor do Conteúdo: Tem como principais funções prover conteúdo para os usuários receptores, assim como um gerenciador de usuários. Utiliza os serviços do provedor da nuvem para armazenar e distribuir conteúdo.
- Usuários Receptores: Caso possua as credenciais necessárias, pode ter acesso ao conteúdo armazenado na nuvem.

O trabalho de Xiong et al. (2012) baseia-se no conceito de "proxy" recifragem, do qual um texto cifrado c_{k1} pode ser transformado em outro texto cifrado c_{k2} com uma chave $rk_{k1 \to k2}$ sem revelar o conteúdo, tal que c_{k1} e c_{k2} só podem ser decifrados por chaves diferentes, como $k1$ e $k2$ respectivamente.

As funções que o trabalho possui são as seguintes:

- Inicialização do Sistema: Criação de chaves do sistema
- Publicação do Conteúdo: Cifrar o conteúdo e publicar na nuvem
- Recuperação de Dados: Deve-se gerar chaves de recifragem, recifrar o conteúdo e decifrar localmente
- Adicionar Novo Membro: Deve-se gerar novas chaves do sistema e redistribuir para todos os usuários

- Remover Membro: Deve-se gerar novas chaves a partir das chaves antigas dos usuários que foram revogados

Esse modelo apresenta as mesmas limitações do trabalho de **??**)ateniese2006improved) , pois ao revogar ou adicionar membros, existe um custoso processo de redistribuição de chaves criptográficas. Caso a dinamicidade do grupo seja alta, a complexidade e a execução de tais processos podem ser inviáveis. Outro ponto negativo é a impossibilidade de se compartilhar um documento com um grupo sem saber ao certo quem são os mesmos. Essa limitação é devido a exigência de se gerar uma chave de recifragem para cada membro existente no grupo. Logo, caso existam M membros, devem existir M chaves de recifragem no sistema para cada grupo, gerando assim também um número muito alto de chaves e tornando o gerenciamento cada vez mais complexo.

4.3 GERENCIADORES LOCAIS DE SIGILO

Existem trabalhos na literatura que propõem o uso de gerenciadores locais para o controle do sigilo de documentos sensíveis. Esses trabalhos tentam lidar com a segurança ponto a ponto, de tal forma que o controle do sigilo não deva ficar nas mãos de uma terceira parte confiável ou no provedor da nuvem. Nesses trabalhos, os usuários cifram localmente os arquivos antes de enviar aos provedores de nuvem. Para isto, estes usuários deverão compartilhar entre si suas chaves criptográficas para que o receptor consiga decifrar.

O trabalho de Pearson, Shen e Mowbray (2009) propõe o uso de um Gerenciador de Privacidade local (GP) que ajudará um usuário a gerenciar de maneira mais eficiente o sigilo de seus dados na nuvem. A ideia da proposta é que os dados sejam cifrados localmente no GP e então enviados para a nuvem cifrados. Os dados serão processados de forma cifrada, resultando em outro dado também cifrado. Os dados serão então enviados para o usuário, que poderá decifrar utilizando seu GP. Desta forma, o trabalho tenta garantir a segurança dos dados não permitindo que o provedor da nuvem tenha contato com o conteúdo em claro.

O trabalho comenta que é inviável aplicações na nuvem trabalharem com todos os dados cifrados. Portanto, deve-se adotar um esquema mais flexível no qual o usuário possa definir quais dados podem ser visualizados pelo provedor da nuvem. Este modelo define dois tipos diferentes de propriedades que o usuário pode configurar: Preferências e Perfil.

Preferências: O modelo propõe o uso de preferências para que o usuário final consiga configurar quais propriedades de compartilhamento es-

tes arquivos vão possuir. Estas propriedades são comunicadas para um controle de acesso que funciona em conjunto com o serviço da nuvem. Estas preferências devem ser associadas a cada dado enviado à nuvem e, preferencialmente, utilizando os métodos de criptografia. Estes métodos podem envolver o cifrамento destas políticas com chaves criptográficas que serão compartilhadas entre os usuários. Para tornar o modelo de criptografia mais confiável, pode-se utilizar técnicas de envelopamento com a criptografia de chaves públicas. Outro modo seria a utilização da criptografia baseada em identidade para se utilizar políticas de sigilo baseada em políticas. Estas políticas poderiam ser utilizadas diretamente como as chaves públicas da criptografia baseada em identidade.

Perfil: A propriedade de perfil permite que o usuário pré-configure e escolha diferentes maneiras de interagir com os serviços de nuvem. Em determinados contextos, o usuário pode querer se comunicar anonimamente, assim como em outros momentos este mesmo usuário pode querer se identificar parcialmente ou totalmente aos serviços utilizados. O uso de perfis provê uma interface simples para a escolha de modalidades diferentes de privacidade com relação a sua identidade e quais dados o provedor da nuvem poderá ter acesso.

A arquitetura do modelo é baseada no uso das seguintes entidades: Usuário, Trusted Platform Module (TPM), Gerenciador de Privacidade e Aplicação na Nuvem.

- Gerenciador de Privacidade: Este gerenciador tem como principal função ajudar o usuário a proteger seus dados sigilosos quando armazenados em provedores de nuvem. A principal característica desse gerenciador é que este realiza as funções de cifрамento e decifрамento sobre os dados, reduzindo assim a quantidade de informação que a nuvem conhece. O gerenciador de privacidade também oferece a configuração sobre os detalhes de compartilhamento e nível de sigilo sobre os dados.

- TPM: O gerenciador de sigilo baseia-se no uso de um hardware seguro chamado “Trusted Platform Module“ (TPM). Este componente, junto com a máquina do usuário, age como raiz de confiança, provendo também mecanismos de criptografia e armazenamento seguro para as chaves criptográficas. Além de proporcionar os mecanismos de sigilo, o TPM também pode prover o mecanismo de integridade do gerenciador de sigilo, garantindo assim mais confiabilidade no sistema com um todo.

O trabalho propõe o uso da criptografia homomórfica para que os dados sejam cifrados e enviados à nuvem. Esses dados cifrados devem poder

ser manipulados pela nuvem para realizar cálculos ou funções sem que o conteúdo dos dados sigilosos sejam revelados. A álgebra responsável pela criptografia homomórfica pode ser descrita da seguinte maneira. Um usuário possui um dado sigiloso x e deseja executar uma função f sobre este dado sem revelar o conteúdo para a aplicação que pode calcular funções $f_1 ... f_n$. Pode-se utilizar a aplicação para calcular uma função f de uma forma sigilosa se para um número positivo m existem funções de ciframento $o_1 ... o_m$ de tal maneira que seja difícil determinar x pelas tuplas $o_1(k,x) ... o_m(k,x)$ sem ter a posse da chave k e de uma função de deciframento d de tal forma que para todas as entradas x e todas as chaves k, $d(k, f_1(o_1(k,x)), ... f_m(o_m(k,x))) = f(x)$.

Para realizar a operação de criptografia homomórfica, primeiro deve-se cifrar o dado x utilizando uma das funções de ciframento para gerar a tupla $(o_1(k,x) ... o_m(k,x))$, utilizando uma chave k que é de custódia do usuário e que fica fora do alcance do provedor da nuvem. Deve-se então enviar estes valores para a aplicação que fará a computação dos valores $f_1(o_1(k,x))$, $... f_m(o_m(k,x))$. Para obter os dados, deve-se utilizar a função de deciframento para obter o valor de $f(x)$ com o auxílio da chave k. Uma vez que o único dado que a aplicação receberá será a tupla $(o_1(k,x) ... o_m(k,x))$, a computação de x torna-se uma tarefa difícil.

Com estas características, este modelo é baseado na segurança ponto a ponto, em que os usuários devem compartilhar chaves para que tenham acesso a conteúdos sigilosos. O provedor da nuvem nunca detém controle das chaves e conteúdos sigilosos, pois executa funções homomórficas para obter novos dados. O trabalho não apresenta uma descrição detalhada sobre o compartilhamento das chaves e como deve ser feito em caso de grupos de usuário compartilhando arquivos sigilosos. A limitação do trabalho está na complexidade envolvida no compartilhamento destas chaves e na revogação dos usuários.

O trabalho de Bessani et al. (2013), mais conhecido como DepSky, propõe uma arquitetura para o armazenamento e compartilhamento seguro de documentos em diferentes provedores de nuvem. Essa arquitetura é baseada no uso de criptografia simétrica, segredo compartilhado de Shamir e *erasure code*. O DepSky tem como principal objetivo o armazenamento seguro e tolerante a falhas bizantinas. O uso das chaves criptográficas torna o trabalho mais completo, pois consegue garantir níveis de segurança pelo uso da criptografia.

O DepSky possui as seguintes entidades: Provedores de Conteúdo, Provedores de Nuvem e Receptores de Conteúdo. Conforme descrito pelo trabalho, as entidades possuem os respectivos objetivos:

- **Provedor de Conteúdo:** São os responsáveis por fornecer dados e enviar para os provedores de nuvem públicas. Devem criar e comparti-

lhar entre si um par de chaves para assinar e verificar a integridade do conteúdo enviado. Devem também criar chaves simétricas para cifrar os dados sigilosos. As chaves simétricas são quebradas utilizando o algoritmo de segredo compartilhado de Shamir e distribuídas nos provedores de nuvem.

- **Provedor de Nuvem:** Entidade responsável por fornecer o serviço de armazenamento e controle de acesso ao conteúdo.
- **Receptor de Conteúdo:** Tem como principal objetivo obter os documentos sigilosos que foram compartilhados pelos provedores de conteúdo e devem ter acesso às chaves simétricas para decifrar os dados obtidos dos provedores de nuvem.

O trabalho menciona dois algoritmos para tratar os desafios encontrados no compartilhamento seguro. O primeiro algoritmo de escrita é composto pelos seguintes passos:

1. Gerar uma chave simétrica k;
2. Cifrar o dado D sigiloso com a chave simétrica k resultando em E;
3. Quebrar a chave k em N partes;
4. Codificar o dado cifrado E em N partes;
5. Quebrar um arquivo em N partes, sendo N o número total de provedores de nuvem;
6. Obter o resumo criptográfico de cada parte $E[N]$ e $k[N]$;
7. Assinar os metadados com uma chave privada Ks, incluindo os resumos criptográficos das partes que foram quebradas e codificadas;
8. Escrever na nuvem com uma versão mais recente do que a já existente.

O primeiro algoritmo consiste em um usuário que tem o direito de escrever os dados na nuvem consiga executar os passos acima em um ambiente seguro para o envio das partes para cada provedor de nuvem pública. O algoritmo prevê que o usuário já tenha previamente compartilhado a chave privada Ks com outros usuários que tem o poder de escrita. O segundo algoritmo de recuperação e leitura dos dados sigilosos dos provedores da nuvem é composto pelos seguintes passos:

1. Recuperar os metadados de cada parte de cada provedor de nuvens m;

2. Recuperar as partes individuais *tmp* de cada nuvem;

3. Conferir se o resumo criptográfico da parte obtida é igual ao obtido junto com os metadados;

4. Ao conferir o número mínimo de partes, deve-se decodificar o arquivo cifrado E;

5. Combina-se as partes da chave simétrica k;

6. Decifra-se o documento cifrado E com a chave simétrica k, obtendo-se D;

Os usuários que tem o direito de escrever os dados na nuvem compartilham entre si chaves privadas para assinar os dados, enquanto que os usuários cujos direitos são apenas de leitura devem compartilhar em si a respectiva chave pública para comparar as assinaturas. O trabalho reutiliza o controle de acesso dos provedores de nuvem para garantir a distribuição correta das chaves, entretanto, não menciona como deve ser o compartilhamento das chaves assimétricas para assinatura.

O trabalho usa o conceito de *erasure code* para diminuir o tamanho das partes dos arquivos que são distribuídas em múltiplos provedores de nuvem. Esse conceito é utilizado neste trabalho de dissertação para otimização do espaço utilizado e diminuição dos custos com os provedores de nuvem. O conceito de quebrar as chaves utilizando segredo compartilhado de Shamir também é utilizado neste trabalho pelo fato de que as chaves possuem pequeno tamanho, logo, mesmo que as partes possuam o mesmo tamanho do arquivo original, o impacto do custo de armazenamento será pequeno.

O sistema DepSky de Bessani et al. (2013) usa conceitos semelhantes que foram utilizados neste trabalho, como a criptografia simétrica, segredo compartilhado de Shamir e *erasure optimal code*. Contudo, o artigo não propõe mecanismos necessários para compartilhar as chaves de forma segura para garantir a integridade das partes dos arquivos sigilosos. O trabalho admite que existe um mecanismo para compartilhamento de chaves assimétricas, no entanto, este é um dos principais desafios para a garantia do sigilo na nuvem utilizando criptografia. Outro ponto é o armazenamento das chaves assimétricas. Como elas são geradas por fora do esquema proposto pelo trabalho, nota-se que a dificuldade no armazenamento e no gerenciamento para grupos continua sendo um desafio, visto que esses são pontos cruciais na segurança de um sistema.

O trabalho de Bessani et al. (2013) possui também como vulnerabilidade a reutilização do controle de acesso dos provedores de nuvem pública para armazenamento, possibilitando assim um conluio entre provedores para

acessar dados de usuários. Uma vez que se obtém acesso aos dados de autenticação do usuário, pode-se obter as partes necessárias para reconstruir as chaves simétricas e dessa forma, decifrar os documentos e ter acesso aos dados confidenciais.

4.4 SIGILO COM CRIPTOGRAFIA BASEADA EM IDENTIDADE

Outros trabalhos que propõem mecanismos para a garantia do sigilo na nuvem utilizam a criptografia baseada em identidade (CBI). Estes trabalhos tem como principal objetivo simplificar o gerenciamento das chaves criptográficas, não utilizando os métodos tradicionais de criptografia de chaves públicas. O CBI tem como principal característica o uso de identificadores conhecidos como chaves públicas. Nesse caso, são utilizados como chaves públicas o e-mail de um usuário, o identificador dentro de uma organização ou até mesmo um conjunto de atributos que este possui. Esta criptografia se comporta diferente da tradicional criptografia de chaves públicas, no qual as chaves são geradas de forma aleatória e então são vinculadas aos dados pessoais de um usuário por meio de certificados.

O trabalho de Yan, Rong e Zhao (2009) baseia-se no uso do CBI, contudo, o mesmo relata limitações no uso da ideia original. Neste modelo, os distribuidores de chaves privadas (DCP) tem a função não só de gerar as chaves privadas, como também autenticar e transmitir as chaves por um canal seguro. Em um sistema em que podem existir milhares de usuários, este DCP pode se tornar um gargalo para as operações de distribuição das chaves.

O trabalho de Yan, Rong e Zhao (2009) propõe o uso da criptografia hierárquica baseada em identidade (CHBI). No modelo de CHBI, existe um DCP raiz que gerará e distribuirá chaves para os DCPs intermediários ou finais. Estes DCP finais irão gerar e distribuir chaves para outros DCP finais ou usuários finais. O uso do CHBI pode diminuir a carga sobre os DCP devido a distribuição das requisições de autenticação, geração e distribuição das chaves privadas.

Os métodos do modelo CHBI são semelhantes ao CBI, entretanto, acrescenta-se os métodos de emissão de chaves para os DCP intermediários ou finais. Conforme o trabalho de Yan, Rong e Zhao (2009), os métodos do CHBI são os seguintes:

- Inicialização do DCP Raiz: O DCP raiz irá inicializar suas variáveis e criará sua chave pública e a chave mestra. Esta chave mestra será utilizada para a geração das chaves dos DCPs intermediários e finais. Os parâmetros públicos são divulgados e utilizados para a geração das chaves públicas dos DCPs intermediários e finais.

- Inicialização dos DCPs Intermediários/Finais: Cada DCP intermediário/final irá receber os parâmetros do DCP raiz e gerará suas próprias chaves mestras. Estas chaves mestras serão utilizadas para a geração das chaves dos usuários finais ou outros DCPs finais.

- Extrair Chave Privada: Quando um usuário ou DCP em um nível t com suas identidades $(ID_1, ...ID_t)$ requisitar suas chaves privadas do DCP nível acima, no qual $(ID_1, ...ID_i)$ são as identidades dos seus antecessores em um nível i $(1 \leq i \leq t)$, o DCP do nível acima irá utilizar esta identidade, os parâmetros públicos do sistema e sua própria chave mestra para gerar as chaves privadas dos usuários.

- Ciframento: Um usuário que queira cifrar uma mensagem M deve utilizar os parâmetros públicos do sistema, a identidade do receptor e a mensagem como parâmetros de entrada para gerar um texto cifrado $C = Cifrar(parametros, ID_{receptor}, M)$.

- Deciframento: Ao receber um texto cifrado, o usuário deve utilizar sua chave privada que recebeu do DCP para tentar decifrar um texto $M = Decifrar(parametros, k, C)$, tal que k é a chave privada do usuário.

O trabalho apresenta problemas da custódia da chave, no qual DCPs tem acesso a todas as chaves dos usuários. Como os DCPs possuem o controle sobre as chaves, podem decifrar e cifrar mensagens em nome dos usuários sem que estes saibam. Desta forma, os DCPs devem ser altamente confiáveis. O trabalho propõe o uso do CHBI para conter a custódia das chaves privadas pelos DCPs, alegando que reduzem o problema a casos mais locais e específicos devido ao uso da hierarquia. Entretanto, o problema continua presente, apresentando os mesmos perigos e problemas de se utilizar um único DCP.

O modelo propõe o uso federado de identidades entre nuvens utilizando-se do CHBI. O trabalho defende a ideia de nuvens híbridas, em que DCPs estarão sob controle de nuvens privadas e públicas. Para que isto ocorra, deverá existir um DCP raiz confiável para todos os servidores de nuvens públicas e privadas. Cada DCP poderá possuir como chave pública o número de identificação de rede (IP) que será subordinado a seus DCP raízes. Desta forma, as chaves públicas serão composições de identificações que remeterão a um caminho confiável até o DCP raiz. Como exemplo, pode-se prover um DCP para uma universidade como a UFSC em que o identificador é $UFSC$ e uma aluna chamada Alice possuirá um identificador $UFSC.Alice$.

Nesse modelo, o autor propõe o uso de dois níveis, o DCP raiz é o $nivel_0$ e os DCPs nas nuvens públicas ou privadas possuem o $nivel_1$. Para realizar a inicialização do sistema utilizando-se o CHBI, deve-se realizar os seguintes passos:

1. O DCP raiz gera dois grupos, um grupo aditivo G_1 de ordem q e um grupo multiplicativo G_2 também de ordem q. Também criará um emparelhamento bilinear admissível $\hat{e}(aP, bQ) = \hat{e}(P,Q) \neq 1 (G_1, G_2, \hat{e}, P_0, Q_0, H_1, H_2)$.

2. O DCP raiz escolhe um $P_0 \in G_1$ e um $s_0 \in Z_q^*$ e atribui $Q_0 = s_0 P_0$.

3. O DCP raiz escolhe as funções de resumo criptográfico $H_1 : 0,1^* \rightarrow G_1$ e $H_2 : G_2 \rightarrow 0,1^n$.

Para gerar uma chave de um DCP, a chave pública pode ser gerada como $P_{UFSC} = H_1(UFSC)$ e sua chave privada será gerada da seguinte forma: $s_{UFSC} = S_0 P_{UFSC}$. Para um usuário Alice, que é de nível inferior ao da Universidade Federal de Santa Catarina (UFSC) como um aluno, cujo identificador pode ser representado por $UFSC.Alice$, a sua chave pública será gerada da seguinte forma: $P_{UFSCAlice=H_1(UFSC||Alice)}$ e sua chave privada será gerada da seguinte forma: $s_{UFSCALICE} = s_{UFSC} + s_{UFSC} P_{UFSCAlice}$.

Esse modelo apresenta limitações com relação a custódia da chave, pois os DCPs continuam tendo acesso às chaves privadas dos usuários. O DCP raiz tem o controle de todos os DCPs intermediários, e desta forma, estes DCPs conseguem ter acesso às chaves dos DCPs intermediários. Outro ponto limitante deste trabalho é a revogação das chaves dos usuários. Ao utilizar identificadores absolutos como nome da entidade e nome do usuário, ao comprometer a chave privada deste, este ficará impossibilitado de receber outra chave privada, a menos que mude de identificador, tornando a solução inviável.

O trabalho de Chow et al. (2012) também utiliza a criptografia baseada em identidade para suas funções de sigilo. O trabalho propõe mecanismos para garantir a segurança e dinamicidade no gerenciamento dos usuários para a garantia do sigilo de documentos eletrônicos compartilhados na nuvem.

O modelo do sistema consiste em três entidades: Provedor de Serviços da Nuvem (PSN), do qual provê o serviço de armazenamento dos arquivos. O Gerente dos Grupos (GG) representando um departamento de tecnologia de uma organização que possui muitos arquivos que devem ser armazenados na nuvem, responsável também pela manutenção dos serviços disponibilizados na nuvem e também gerencia as credenciais dos usuários. E por fim, os Usuários (U) que representam os funcionários de organizações que enviam e recuperam arquivos dos serviços da nuvem.

Os métodos que o trabalho utiliza para o gerenciamento das chaves são listados a seguir:

- Inicialização: Esta fase é realizada pelo GG e recebe como entrada um parâmetro de segurança e que retorna parâmetros públicos PK e uma

chave mestra secreta MK. Os parâmetros PK são então divulgados e a chave MK é armazenada de maneira segura pelo GG.

Ao receber um parâmetro de segurança k, o GG executa os seguintes passos:

1. Gera um sistema de grupo bilinear $(p, G_1, G_2, G_T, F(.), e(.,.))$, no qual G_1, G_2 e G_T são grupos cíclicos multiplicativos de ordem p, $F : G_2 \rightarrow G_1$ é um isomorfismo e $e : G_1 \times G_2 \rightarrow G_T$ é um emparelhamento bilinear admissível. Com $p \geq 2^k$ e com as funções de resumo criptográfico $H_0 : 0,1^* \rightarrow Z_p^*$, $H_1 : 0,1^* \rightarrow G_2^2$, $H_2 : 0,1^* \rightarrow Z_p$.
2. Escolhe aleatoriamente um $h_1 \in G_1$, $h_2, g_2 \in G_2$ e computa $g_1 = F(g_2)$.
3. Aleatoriamente escolhe dois segredos $K, L \in Z_p^*$ e atribui $w_1 = h_1^K$, $w_2 = g_2^L$.

- Registro do Usuário: Esta é uma fase inicial de registros de usuários no sistema, realizado pelo GG e usuários. Um usuário com identificador U_{ID} pode se cadastrar com o GG e receber um par de chaves de decifração (ak_{ID}, dk_{ID}). Além disso, o ID é adicionado à uma lista de usuários S do qual é publicado e utilizado para o ciframento *broadcast*.

  Para cada usuário U_{ID}, o GG emite uma chave privada. A identidade ID é adicionada à lista S e $(ID, g_1^{1/(L+x_{ID})})$ é adicionada à lista de tokens T.

- Acesso aos Dados: Esta fase é realizada por um usuário U_{ID} que possui o par de chaves e o PSN. Um usuário pode acessar os dados cifrados da nuvem e decifrar utilizando a chave privada. Para enviar os dados, o usuário irá assinar anonimamente o arquivo com a chave privada. O PSN apenas aceitará o arquivo assinado caso a assinatura seja válida e gerada por um usuário não revogado.

- Inclusão de Usuário: Esta fase é realizada pelo GG e um novo usuário U_{ID}. O GG primeiramente envia o par de chaves ao usuário e então adiciona o identificador na lista S. O GG então decifra os textos cifrados existentes para que o novo usuário possa decifrá-los e ter acesso aos dados.

- Revogação de Usuário: Esta fase é realizada pelo GG e pelo provedor da nuvem. Caso um usuário U_{ID} seja revogado, o GG remove o identificador ID da lista S. Então o GG procura pelo par $(ID, y_I D)$ na lista de

tokens T e envia $y_I D$ para o provedor de nuvem de tal forma que este adicione na lista de revogação R.

O trabalho possui sérias limitações com relação a custódia das chaves. Como existirá um administrador do sistema que tem acesso às chaves, desde o momento da criação até a distribuição, existe um risco de mau uso das chaves criptográficas. O trabalho propõe a renovação das chaves e redistribuição a cada mudança dos grupos. Este processo pode ser inviável caso os grupos sejam muito dinâmicos e com um número elevado de membros. Outro limitante da proposta são os tamanhos dos textos cifrados que mudam conforme o tamanho dos grupos. Como o texto cifrado é resultante de operações que utilizam os identificadores de cada membro, caso o grupo possua um elevado número de membros, o texto cifrado poderá ter um tamanho grande, inviabilizando o armazenamento destas chaves.

Os trabalho de Ruj, Nayak e Stojmenovic (2011), Jung et al. (2013), Emura et al. (2009) e Kamara e Lauter (2010) utilizam o conceito de criptografia baseada em atributos CBA, no qual as chaves estão vinculadas matematicamente aos atributos de usuários. Esse vínculo se dá pelas operações de transformação de atributos em modelos matemáticos para operações e verificações de usuários. Os modelos dos sistemas baseiam-se no uso de múltiplos distribuidores de chaves privadas DCP e apenas um provedor de nuvem para o armazenamento dos documentos cifrados.

Os métodos utilizados no esquema são:

- **Inicialização do Sistema:** A inicialização do sistema é feita de maneira distribuída, fazendo com que cada DCP $A_j \in A$ possua um conjunto de atributos L_j que devem ser distribuídos aos usuários. Cada DCP possui um conjunto diferente de atributos, de tal forma que $L_i \neq L_j$ para todo $i \neq j$.

- **Geração das Chaves:** Um usuário deve entrar em contato com cada DCP para obter um conjunto de atributos e uma chave secreta correspondente. O usuário possuirá um conjunto de chaves secretas para cada DCP e parâmetros públicos para realizar o ciframento de dados.

- **Ciframento:** Para realizar o ciframento de um dado sigiloso, o emissor da mensagem deve primeiramente definir quais serão os atributos utilizados para limitar o acesso aos documentos. Dentre esses atributos, cria-se um subconjunto mínimo necessário para o deciframento. Após a definição dos atributos, o usuário irá utilizar os parâmetros públicos dos DCPs como chave pública para o ciframento final da mensagem.

- **Deciframento:** O deciframento de uma mensagem é realizado utilizando-se os atributos necessários, a chave secreta e os parâmetros públicos.

Este deciframento só será possível caso os atributos satisfaçam ao menos um subconjunto mínimo definido no momento do ciframento. Caso o usuário possua o número mínimo de atributos necessário e as chaves secretas necessárias dos DCPs, o deciframento da mensagem será realizado com sucesso.

- **Revogação:** Para que a revogação seja efetuada, os dados cifrados devem ser recifrados, pois os atributos e chaves utilizadas são as mesmas para diferentes arquivos. Ou seja, o conjunto de atributos L que um usuário U possui é sinalizado como revogado e todos os usuários devem mudar os dados armazenados que possuem atributos $l \in L$.

Esses trabalhos que utilizam a criptografia baseada em atributo possuem em comum o problema de revogação, pois ao enviar atributos a um usuário, estes deverão ser renovados caso queiram ter revogação de arquivos para os mesmos tipos de atributos. A complexidade e a quantidade de passos na comunicação envolvida para tais processos é outro ponto negativo. O processo de revogação torna-se lento e exige o recifrамento de documentos, devido a utilização de atributos enviados aos usuários. Dessa forma, os DCPs ou usuários terão de se preocupar em recifrar documentos para garantir a revogação de usuários. Outro ponto negativo é a falta de tolerância a falhas utilizando DCPs que possuem atributos únicos. Ou seja, caso um desses DCPs fique indisponível, os atributos não poderão ser disponibilizado para os usuários que, consequentemente, não poderão cifrar ou decifrar dados.

O trabalho de Zhou, Varadharajan e Hitchens (2011) utiliza conceitos semelhantes ao proposto neste trabalho de mestrado, no qual utiliza criptografia baseada em papéis. Esse conceito é baseado no uso da criptografia baseada em identidade, modificando-se matematicamente para atender aos requisitos impostos por Zhou. O trabalho consiste das seguintes entidades:

- Um conjunto de donos de documentos que desejam compartilhá-los na nuvem;
- Um conjunto de usuários que donos de arquivos desejam compartilhar documentos;
- Um conjunto de papéis que os usuários podem possuir;
- Gerentes de Papéis (GP);
- Administrador de Grupos (AG);

O trabalho assume que existe apenas uma autoridade Administradora de Grupos que gera chaves para usuários e papéis. O esquema é definido pelos seguintes procedimentos para a criptografia baseada em papéis:

- **Inicialização do Sistema:** Toma como entrada um parâmetro de segurança λ e tem como saída uma chave mestra *mk* e uma chave pública de grupo *pk*. A chave mestra *mk* é dada ao AG e a chave *pk* é divulgada.

- **Criar Papel:** Algoritmo executado pelo AG, ao receber como parâmetro um papel cujo identificador é ID_R gera o segredo sk_R do papel e retorna um conjunto de parâmetros públicos pub_R do papel, assim como uma lista vazia de usuários *LU* que listará todos os usuários que fazem parte desse papel.

- **Criar Usuário:** Algoritmo executado pelo AG que ao receber como parâmetro a identidade ID_U de um usuário retorna a chave de decifração *dk*.

- **Adicionar Usuário:** Algoritmo executado pelo GP que ao ser requisitado para entrar em papel por um usuário, atualiza os parâmetros públicos pub_R e a lista de usuários *LU* caso o usuário satisfaça os requisitos necessários para o papel.

- **Cifrar:** Algoritmo executado pelo dono do arquivo que deseja cifrar um dado *M* que retorna um dado cifrado *C* que é armazenado na nuvem.

- **Decifrar:** Algoritmo executado por um usuário para decifrar um dado cifrado na nuvem retorna o dado em claro *M* caso o usuário possua permissões para acessar o dado.

- **Revogar Usuário:** Algoritmo executado pelo GP e AG que ao receber um identificador ID_U de um usuário U, remove U da lista de usuários *LU* e atualiza os parâmetros públicos.

A proposta de Zhou, Varadharajan e Hitchens (2011) é eficiente apenas quando não há muitas revogações, caso contrário, apresentará alta complexidade devido ao fato de que toda vez que há uma revogação de usuário, deve-se gerar novos parâmetros públicos. O que significa que mensagens cifradas antes de executar a função de revogação não podem ser decifradas pelos usuários restantes do grupo pelo fato de que o parâmetro público foi atualizado. Para evitar esse problema, deve-se criar índices a cada vez que houver alguma revogação. Dessa forma, o provedor de nuvem verifica qual índice dos parâmetros público é necessário para decifrar o documento. Outro ponto negativo na revogação é que em todos os procedimentos de revogação, deve-se atualizar parâmetros de segurança de todos os papéis do sistema. Isso torna o sistema complexo e ineficiente caso possua muitos papéis. O trabalho de Zhou não resolve o problema das custódias das chaves (*key escrow*). O administrador do sistema tem acesso às chaves privadas no método

de extração. Outro problema é o ponto único de falha, fazendo com que caso o administrador seja comprometido, parte do sistema de revogação também seja comprometida.

4.5 CONCLUSÃO DO CAPÍTULO

Os trabalhos referenciados foram divididos em três grupos para realizar uma melhor análise de cada tipo de proposta. Os trabalhos que focam na proposta de sigilo como serviço suprem necessidades de compartilhamento de documentos sigilosos, contudo, necessitam de um mecanismo complexo para o gerenciamento das chaves criptográficas. Outra limitação encontrada nesses trabalhos está relacionada com a revogação de um usuário que pode ser muito custosa devido ao reciframento de grande parte dos arquivos. Isso se deve ao fato de que não existe um mecanismo eficiente para revogar o acesso aos usuários a determinados documentos e chaves. Outro ponto negativo é a baixa granularidade das chaves criptográficas. Em uma exposição indevida das chaves criptográficas, compromete-se grande parte da segurança do sistema, enquanto que o ideal seria comprometer apenas uma pequena parte, como apenas um documento.

Os trabalhos cuja ideia foi centralizada no uso de gerenciadores locais de sigilo possuem limitações com relação ao compartilhamento de mensagens e dados cifrados. Uma delas é a falta de mecanismos eficientes de revogação. Como algumas chaves de sigilo são compartilhadas entre usuários e não possuem granularidade suficiente, o comprometimento de chaves pode ocasionar problemas na segurança do sistema. Outro ponto fundamental é o gerenciamento das chaves, cuja tolerância a falhas não é mencionada em alguns trabalhos. Ou seja, na ausência de um servidor fornecedor das chaves, os usuários não poderão decifrar ou cifrar dados sigilosos. Outro ponto negativo é a confiança no controle de acesso dos provedores de nuvens públicas, tornando o modelo proposto a vulnerabilidades em caso de conluio entre os provedores de nuvem de armazenamento.

Os trabalhos focados na utilização da criptografia baseada em identidade e atributos possuem diferentes limitações. Enquanto alguns abordam tópicos como um melhor gerenciamento das chaves utilizando atributos ou baseado em papéis, outros possuem limitações com relação a revogação e granularidade das chaves criptográficas. Outros problemas encontrados também foram as faltas de tolerância a falha. Alguns trabalhos propõem o uso distribuído de gerenciadores de chaves, entretanto, existem restrições com relação ao uso de atributos na indisponibilidade de gerenciadores devido a falhas.

Ao final, conclui-se que muitos desses trabalhos propõem mecanismos eficientes e que conseguem em partes prover o sigilo necessário para o compartilhamento de documentos sigilosos. Entretanto, existem muitas limitações com relação a efetiva segurança no gerenciamento das chaves criptográficas. Outro ponto importante a se destacar é que poucos trabalhos relacionados comenta sobre a tolerância a falhas no armazenamento desses documentos sigilosos. Nesses casos, os trabalhos utilizam normalmente apenas um provedor de nuvem para todo o armazenamento. Caso esse provedor de nuvem venha a falhar, os usuários não poderão ter acesso aos seus dados. Dessa forma, faz-se necessário o uso de forma distribuída de diferentes provedores de nuvem para uma melhor garantia no fornecimento dos dados dos usuários. A Tabela 1 demonstra um comparativo das propriedades alcançadas entre os trabalhos relacionados.

Devido a falta de soluções que contemplem todas as propriedades necessárias, uma arquitetura de middleware se faz necessária para unir todos os aspectos importantes para a segurança no compartilhamento de documentos sigilosos em nuvem. Dentre esses aspectos, pode-se destacar: gerenciamento seguro de chaves criptográficas, tolerância a falhas e baixa complexidade nas interfaces de acesso às funções.

Procedimento	Gerenciamento Seguro das Chaves	Revogação de Usuários	Tolerância a Falhas
(ATENIESE et al., 2006)	✓	✓	X
(XIONG et al., 2012)	✓	✓	X
(PEARSON; SHEN; MOWBRAY, 2009)	✓	X	X
(BESSANI et al., 2013)	X	✓	✓
(YAN; RONG; ZHAO, 2009)	X	X	X
(CHOW et al., 2012)	X	✓	X
(RUJ; NAYAK; STOJMENOVIC, 2011)	✓	X	✓
(JUNG et al., 2013)	✓	X	✓
(EMURA et al., 2009)	✓	X	✓
(KAMARA; LAUTER, 2010)	✓	X	✓
(ZHOU; VARADHARAJAN; HITCHENS, 2011)	X	✓	X

Tabela 1 – Comparação entre soluções alcançadas de todos os trabalhos relacionados.

5 PROPOSTA DE MIDDLEWARE

5.1 INTRODUÇÃO

Este capítulo tem como objetivo apresentar uma proposta de arquitetura de um middleware para o compartilhamento de documentos sigilosos em nuvem conforme os requisitos apresentados nas seções anteriores. Esses requisitos são organizados e listados para uma melhor compreensão do real problema no compartilhamento de documentos sigilosos.

Partindo-se dos requisitos listados, premissas são definidas para a arquitetura do sistema. Após essas definições, um modelo geral e uma arquitetura mais detalhada são apresentados para um melhor entendimento.

Diagramas de classes ilustram como o middleware foram projetados para uma futura implementação completa. Interfaces de programação de aplicação (*Application Programming Interface* - API) externas e internas do middleware, assim como do módulo de segurança criptográfico, são apresentados em conjunto com seus diagramas de sequência para detalhar o funcionamento do fluxo dos dados.

A partir da arquitetura, algoritmos são apresentados para os principais processos de compartilhamento de documentos sigilosos. A partir dos algoritmos, implementações foram realizadas para validar principalmente a parte de gerenciamento de chaves criptográficas e o uso da criptografia baseada em identidade. A partir dessa implementação, avaliações foram feitas na seção seguinte.

5.2 REQUISITOS DE SEGURANÇA

Baseado na pesquisa realizada, pode-se estabelecer os principais requisitos para elaborar uma arquitetura de middleware para garantir o compartilhamento sigiloso de documentos. Os requisitos necessários para o gerenciamento das chaves criptográficas são:

- **Custódia das Chaves:** Apenas o usuário deverá ter posse da sua chave, evitando assim que as entidades distribuidoras de chaves possam ter posse das mesmas.

- **Forward Secrecy:** Um usuário que foi retirado de um grupo não pode ter acesso aos documentos compartilhados para o grupo após a sua saída.

- **Backward Secrecy:** Um usuário que foi incluído em um grupo não pode ter acesso aos documentos compartilhados antes de sua entrada.
- **Revogação do Acesso:** Após a retirada de um usuário de um grupo, o mesmo não deve mais ter acesso aos documentos cifrados e não poderá decifrar novos documentos que foram compartilhados.
- **Tolerância a Falhas:** O sistema deve fornecer dois níveis de tolerância a falha. Tanto para o gerenciamento de chaves quanto o armazenamento de documentos cifrados. Caso um servidor fique indisponível, deve-se prover mecanismos de contingência dos servidores para manter a solução em funcionamento.
- **Armazenamento das Chaves:** O usuário não deve necessitar armazenar as chaves privadas para realizar futuros procedimentos de criptografia. Ou seja, as chaves devem ser emitidas conforme a demanda.
- **Controle de Acesso:** Os provedores de nuvem não devem ter acesso aos dados dos usuários, como senhas ou outros parâmetros utilizados para autenticação em seus sistemas.

Esses requisitos estão melhores definidos na Seção 3.4 e as ameaças estão descritas na Seção 3.7.5. O sistema de compartilhamento de documentos tem como parte crítica o gerenciamento de chaves criptográficas, contudo, requisitos como disponibilização das partes cifradas e uma melhor distribuição das partes é igualmente importante.

Para garantir a segurança no armazenamento dos documentos e prover uma maior confiabilidade no sistema, deve-se empregar um esquema de distribuição dos arquivos cifrados. Dessa forma, um dos requisitos com relação ao armazenamento baseia-se no uso de diferentes provedores de nuvem, aproveitando a independência dos serviços oferecidos e uma maior probabilidade de disponibilidade dos servidores. Deve-se prover também um mecanismo de tolerância a falhas para o armazenamento dos arquivos cifrados.

5.3 PREMISSAS

Este trabalho tem como principal foco propor uma arquitetura de middleware para prover um compartilhamento seguro de documentos sigilosos. Não será discutido em detalhes o controle de acesso do sistema. Esta proposta segue as seguintes premissas:

- O controle de acesso é fornecido por um sistema de gestão de identidades e quem fornece as credenciais para acessar os provedores de nuvem

são os módulos de segurança criptográficos;

- Não existe concorrência para o acesso de escrita aos documentos;
- Não existe concorrência para realizar alterações no controle de acesso;
- Os algoritmos de criptografia são resistentes a colisão;
- Os provedores de nuvem são semi-confiáveis (Estes irão se comportar corretamente perante as requisições dos usuários, mas são curiosos para ver os dados armazenados);
- Existe um versionamento dos documentos que são escritos na nuvem;
- Módulos de Segurança Criptográficos são confiáveis e possuem mecanismos de segurança física e lógica;
- Os provedores de nuvem públicas e os módulos de segurança criptográficos estão previamente configurados;
- Os canais de comunicação entre os módulos do middleware utilizam túneis seguros, como SSL/TLS.

Por meio dessas premissas limita-se os problemas encontrados no compartilhamento de documentos sigilosos e consegue-se estabelecer um foco maior nas questões mais críticas, como o gerenciamento das chaves criptográficas e o armazenamento dos documentos sigilosos.

5.4 MODELO

O modelo de uso dessa proposta possui quatro componentes principais. O primeiro representa provedores de nuvem que armazenam documentos sensíveis. O segundo são os usuários finais que possuem aplicações e que utilizam o middleware para cifrar e decifrar dados sensíveis, assim como armazenar de forma segura as chaves criptográficas utilizadas. O terceiro componente representa os distribuidores de chaves privadas, embarcados em diferentes MSCs para gerenciar o controle das chaves. E por fim, o quarto elemento é um sistema de gestão de identidades, responsável por verificar a autenticidade do usuário por meio dos DCPs. A Figura 2 mostra uma visão geral do modelo.

- Gerenciadores de Chaves Criptográficas: Estarão embarcados em diferentes MSCs em uma nuvem privada, cujo ambientes são controlados

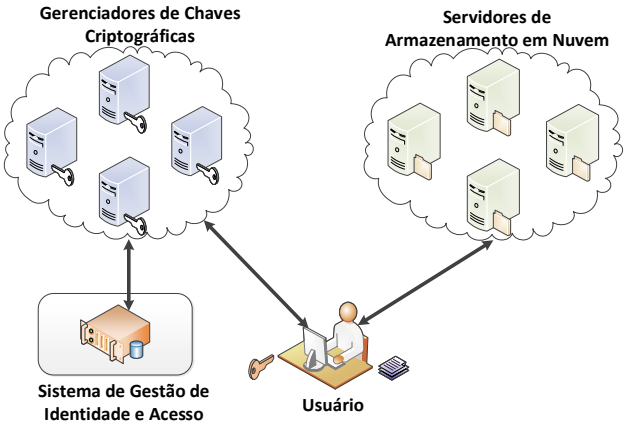

Figura 2 – Modelo Geral de Compartilhamento de Documentos Sigilosos.

fisicamente e com acesso à Internet. Deve-se empregar uma distribuição geográfica, fazendo com que os MSCs possam ser acessados por meio da Internet mas que estejam longe o suficiente para não serem comprometidos facilmente de forma física. Esse gerenciamento utilizará conceitos de criptografia baseada em identidade em conjunto com um rígido controle de acesso. O controle de acesso para o gerenciamento de chaves funciona de forma integrada a outro servidor de gerenciamento de identidades. O gerenciador de chaves criptográficas tem como principal objetivo a correta e segura distribuição das chaves criptográficas para os usuários.

- Provedor de Nuvem Pública para Armazenamento: Os servidores de armazenamento precisam estar hospedados em diferentes provedores de nuvem. As principais características necessárias são: a distribuição do controle de acesso por meio de replicação de estados e o funcionamento distribuído nos nodos dos servidores de aplicação. Os provedores de nuvem são responsáveis pelo armazenamento dos documentos e pelo controle de acesso aos dados. O gerenciador de chaves criptográficas tem como principal objetivo a correta e segura distribuição das chaves criptográficas para os usuários.

- Usuário: O lado do usuário é responsável por editar, cifrar e decifrar arquivos. Os usuários possuem uma aplicação local que interage com

o middleware para realizar as funções criptográficas e funções de compartilhamento dos arquivos sigilosos. Os mesmos definem quem serão os custodiantes dos documentos sensíveis cifrados. Esses custodiantes são delimitados por regras de acesso específicas de cada aplicação.

- Sistema de Gestão de Identidades: Para que os dados de autenticação dos usuários não sejam expostos para os provedores de nuvem, reutiliza-se um sistema de gestão de identidades de uma entidade que o usuário pertence. Dessa forma, o usuário irá se autenticar perante os gerenciadores de chaves criptográficas, que por sua vez, irão consultar o sistema de gestão de identidades para conferir a autenticação. Caso seja bem sucedida, o usuário irá receber um token específico para cada tipo de provedor de nuvem para acessar os dados. Esses tokens fornecem duração e direitos limitados para os usuários que forem compartilhar documentos sigilosos.

5.4.1 Gerenciadores de Chaves Criptográficas

A Figura 3 ilustra o funcionamento do gerenciamento de chaves criptográficas do modelo. Os gerenciadores de chaves criptográficas (GCC) são distribuídos geograficamente e são independentes de funcionamento. Após o procedimento de inicialização, conforme descrito na Seção 3.6.3, cada GCC possuirá uma parte da chave mestre que será utilizada para gerar as chaves privadas dos usuários. Os interessados em recuperar as chaves privadas devem se autenticar perante os GCCs, que obtem as credenciais por meio de um sistema de gestão de identidades independente, e obter o mínimo das partes necessárias para a recuperação da chave, como descrito na seção 3.6.4. Esse procedimento fará com que apenas os usuários que conseguirem se autenticar perante os GCCs de maneira correta consigam obter a chave privada correta ao identificador requisitado.

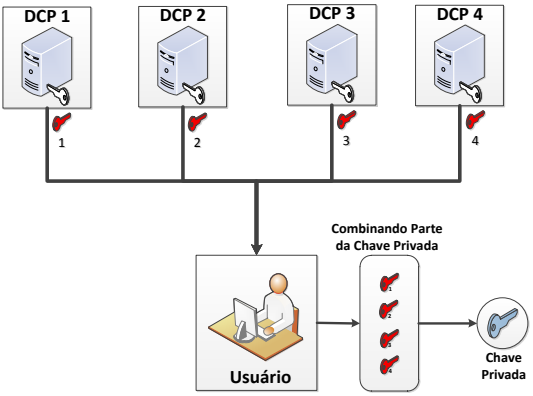

Figura 3 – Recuperação das chaves privadas por meio das multi autoridades.

5.4.2 Provedor de Nuvem Pública para Armazenamento

A Figura 4 ilustra como os procedimentos de armazenamento com sigilo funcionam. Usuários (Provedores de Conteúdo) que desejam compartilhar documentos confidenciais devem primeiramente se autenticar perante os GCCs. Esses irão fornecer um token de autenticação para que os usuários possam depois armazenar os arquivos sigilosos. Após o processo de autenticação, o usuário deve utilizar o middleware para cifrar os documentos localmente, realizar operações de quebra e codificação para então enviar e armazenar os documentos nos provedores de nuvem, juntamente com o token de autenticação para liberar a operação. O provedor de conteúdo deve informar quem terá acesso aos dados sensíveis. O usuário (Receptor de Conteúdo) que deseja obter acesso ao documento sigiloso deve utilizar o middleware para se autenticar perante os provedores de nuvem e obter o dado requerido para montar, decodificar e decifrar o documento. Neste trabalho, serão utilizados abstrações chamadas DataBlocks para armazenar dados dos documentos sensíveis. O DataBlock contém cinco itens: um identificador *ID*, uma parte da chave simétrica, um resumo criptográfico assinado da parte da chave simétrica, uma parte do documento cifrado e um resumo criptográfico assinado da parte do documento cifrado.

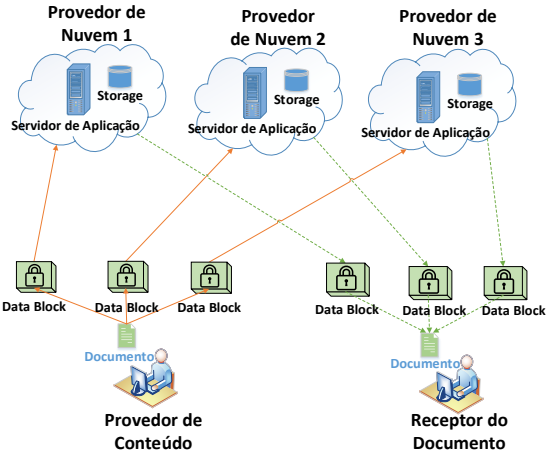

Figura 4 – Criação dos DataBlocks e o compartilhamento utilizando provedores de nuvens públicas.

5.4.3 Sistema de Gestão de Identidades

A Figura 5 ilustra o procedimento de autenticação do usuário por meio do sistema de gestão de identidades. Para que um usuário possa ter acesso ao provedor de nuvem pública para armazenamento, o mesmo já deve ter sido cadastrado em um sistema de gestão de identidades (SGI) em uma aplicação separada. Nesse caso, os mesmos dados para autenticação serão utilizados perante os gerenciadores de chaves criptográficas (GCC). Esses irão receber os dados dos usuários (número 1 da figura) e entrarão em contato com um SGI para conferir se os dados conferem (número 2 da figura). Caso os dados estejam corretos, o SGI informará o GCC que o usuário autenticou-se corretamente e informará quais dados pode ter acesso (número 3 da figura). O GCC então irá entrar em contato com os provedores de nuvem pública para solicitar tokens de acesso para o usuário com as restrições necessárias (número 4 da figura). Os provedores de nuvem para armazenamento retornam tokens de acesso com as devidas restrições e um tempo de expiração (número 5 da figura). Após receber os tokens, o GCC então enviará os mesmos para o usuário que o utilizará para ter acesso por um tempo pré-determinado de uso e com direitos limitados para acessar apenas os dados que pode ter acesso (número 6 da figura). O usuário então utilizará esse token para autenticar-se perante os provedores de nuvem para armazenamento e assim conseguirá ter

acesso aos dados (número 7 da figura).

O SGI é compartilhado entre todos os GCCs, fazendo com que o controle de acesso não precise ser replicado. Após a autenticação, cada GCC irá emitir todos os tokens necessários para acessar os provedores de armazenamento em nuvem. Dessa forma, caso o usuário consiga receber corretamente os N tokens corretamente do primeiro GCC, não necessita mais receber tokens dos outros GCCs para se autenticar perante os provedores de nuvem para armazenamento.

Cada provedor de nuvem possui uma peculiaridade com relação aos tokens, portanto, o GCC deverá emitir tokens conforme as propriedades necessárias para cada provedor de nuvem. Por exemplo, o provedor de nuvem para armazenamendo da empresa Google utiliza o mecanismo *OAuth 2* para a emissão do token de autorização. Assim como a empresa Amazon com seu serviço de armazenamento utiliza conceitos de token com o nome *Token Vending Machine*.

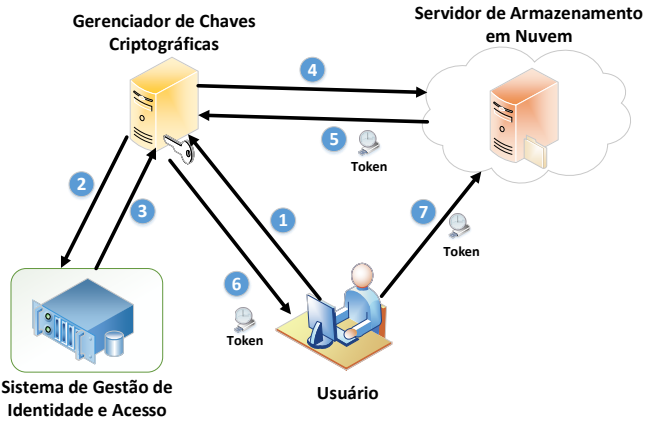

Figura 5 – Processo de autenticação utilizando um sistema de gestão de identidades.

5.5 ARQUITETURA

O middleware de sigilo em nuvem é uma entidade intermediária entre a aplicação do usuário, provedores de nuvem públicas e MSCs em nuvens privadas. O middleware ficará encarregado de todo o processo de comuni-

cação com as nuvens públicas e privadas, possuindo diferentes interfaces de comunicação internas para cada nuvem, podendo montar o tipo de arquivo que cada uma suporta. O middleware também ficará encarregado das principais funções criptográficas, como criação de chaves simétricas, recomposição de chaves baseadas em identidade, assim como funções de cifrar, decifrar, assinar e verificar dados.

A arquitetura possui diferentes módulos para a integração de serviços para aplicações que desejam proporcionar o sigilo no compartilhamento de documentos sensíveis. Os módulos que ficam no cliente são os seguintes:

- **Controller:** Esse módulo será o responsável por disponibilizar uma interface de programação de aplicativos (*Application Programming Interface* - API) para o usuário, afim de que seja feita uma integração entre os serviços de armazenamento em nuvem, gerenciamento de chaves nos módulos de segurança criptográficos e serviços de criptografia.

- **Crypto Handler:** Módulo responsável pelas chamadas que envolvem criptografia, como a criação de chaves simétricas e a construção das chaves assimétricas baseada em identidade, assim como funções que envolver processos de transformação dos dados em claro em dados criptografados. Esse módulo também é responsável por criar chaves simétricas para sigilo localmente e chaves assimétricas baseadas em identidade por meio do módulo HSM API.

  - **Crypto Functions:** Esse módulo é responsável por todas as operações criptográficas que vão envolver dados sensíveis, como arquivos sigilosos e chaves. Esse módulo possui as seguintes funções: Cifrar, Decifrar, Assinar e Verificar.
  - **HSM APIs:** Disponibiliza chamadas para enviar mensagens aos módulos de segurança criptográficos para a criação e recuperação de chaves criptográficas assimétricas baseadas em identidade. Além dessa parte de criação e recuperação, também disponibiliza um método para autenticação perante os módulos de segurança criptográfico para garantir a segurança nas operações. Esse módulo se encarrega de enviar as mensagens nos formatos corretos de cada módulo de segurança criptográfico pré-cadastrado.

- **Storage Handler:** Módulo responsável pelas chamadas que envolvem o armazenamento e recuperação de arquivos sigilosos e que estão cifrados e assinados pelo módulo Key Handler.

  - **Split Functions:** Disponibiliza chamadas para realizar o particionamento dos arquivos e chaves criptográficas envolvidas no

processo de sigilo. O módulo disponibilizará métodos para separar e juntar as partes dos arquivos utilizando o método *optimal erasure code* e chaves com o método de segredo compartilhado de Shamir.

– **Cloud APIs:** Disponibiliza para o File Handler chamadas para enviar mensagens para os provedores de nuvens públicas de armazenamento para autenticação de usuários, armazenamento e busca de arquivos cifrados. Esse módulo é encarregado de enviar as mensagens com os Blocos de Dados nos formatos corretos de cada provedor de nuvem pré-cadastrado.

Os módulos que estão presentes nos módulos de segurança criptográficos, representando a parte do servidor, são:

- **Initialization Functions:** Disponibiliza funções para a inicialização dos módulos de segurança criptográficos. Esses métodos são encarregados de realizar os processos comentados na sub-seção 3.6.3.

- **Key Recovery Functions:** Módulo responsável pelas chamadas que as aplicações fazem para obter as partes das chaves privadas utilizadas nos procedimentos de sigilo. Cada MSC deve ser responsável por suas partes e realizar procedimentos independentes de criação e distribuição das partes.

Parte da solução da arquitetura está localizada em nuvem pública e outra parte fica localizada em nuvem privada. Nessa arquitetura, os gerenciadores de chaves criptográficas são implantados em nuvem privada, utilizando mecanismos de segurança física como hardwares seguros para uma proteção mais efetiva contra ataques internos e de terceiros. O armazenamento de documentos é implantado em provedores de nuvem pública para a melhor utilização da tecnologia, garantindo assim uma maior escalabilidade e economia no investimento de infraestrutura. A Figura 6 ilustra a arquitetura do middleware.

Para a proteção dos distribuidores de chaves, deve-se utilizar MSCs modificados, com a intenção de fornecer proteções físicas e lógicas contra agentes maliciosos internos e externos, conforme descrito na Seção 3.5. Esses módulos funcionam como um computador seguro de propósito geral e garantem a segurança interna utilizando sensores para a detecção de intrusões físicas e utilizando interfaces de comunicação simples para a garantia da segurança lógica do software (SMITH; WEINGART, 1999).

Essa arquitetura prevê que toda a infraestrutura de provedores de nuvens públicas para armazenamento e os módulos de segurança criptográfi-

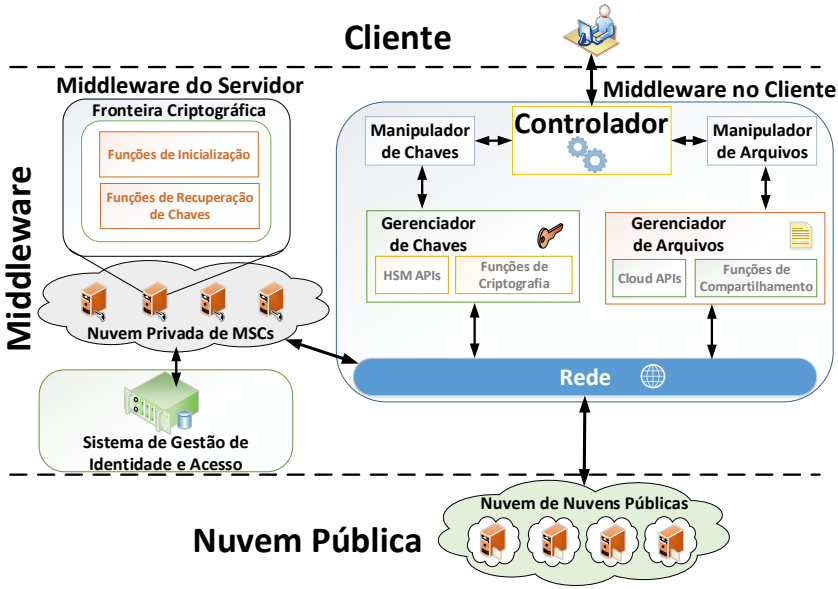

Figura 6 – Arquitetura do Middleware de Privacidade em Nuvem.

cos estejam previamente configurados e com seus controles de acesso pré-definidos. O middleware ficará encarregado de fazer toda a integração entre os processos de autenticação, armazenamento e sigilo, fazendo com que todo o procedimento de compartilhamento de documentos sigilosos seja o mais seguro e transparente possível para o usuário.

5.6 IMPLEMENTAÇÃO

5.6.1 Interface de Programação de Aplicação do Middleware

O middleware oferece funcionalidades para compartilhar e recuperar dados confidenciais entre grupos de usuários utilizando provedores de nuvens públicas para armazenamento. Algumas dessas funcionalidades tem como principais características serem reutilizáveis e configuráveis para o desenvolvimento mais rápido de sistemas que necessitem de compartilhamento de documentos sigilosos. A Figura 7 ilustra o diagrama com as principais classes referente a essa arquitetura. As classes *Storage Handler* e *Crypto Handler*

são as principais classes de controle do middleware e distribuem as requisições conforme forem solicitados pela classe *Controller*. A Figura 8 ilustra o diagrama de classes do módulo *Storage Handler*, contendo informações a respeito dos métodos e tipos utilizados para realizar as funções que interagem com os provedores de nuvens públicas. A Figura 9 ilustra o diagrama de classes do módulo *Crypto Handler*, contendo informações a respeito dos métodos e tipos utilizados para realizar o gerenciamento das chaves, assim como a utilização de funções criptográficas.

Para a realização dos procedimentos de compartilhamento de documentos sigilosos, a arquitetura provê para as aplicações as funcionalidades de armazenamento e criptografia por meio das seguintes Interface de Programação de Aplicação (*Application Programming Interface* - API) APIs:

- **addProvider**$(ipProvider, tipoProvider)$**:** Tem como objetivo adicionar novos provedores de nuvem ao middleware. Essa função é limitada ao número de APIs implementadas pelo middleware. No caso desse trabalho, limita-se em 4. Portanto, pode-se adicionar até quatro diferentes provedores de nuvem no middleware. Essa API necessita receber um conjunto de caracteres contendo o endereço IP do provedor de nuvem pública e o tipo enumerador de nuvem que está implementado no middleware. No middleware desse trabalho existem os enumeradores: Amazon, Azure, Google e Rackspace. Esses tipos foram escolhidos de forma arbitrária, pois não faz parte do foco do trabalho.

- **createHSMUser**$(usuario, senha)$**:** Configura-se o usuário e senha que deve-se utilizar para o gerenciamento de chaves criptográficas nos diferentes MSCs que estão localizados na nuvem privada. Esse usuário deve ser referente ao cadastrado anteriormente no sistema de gestão de identidades. Após essas atribuições, o usuário é autenticado perante os MSCs e o mesmo retorna tokens de autenticação para utilizar com os provedores de nuvem de armazenamento.

- **returnCloudProviders:** Recebe um conjunto do tipo CloudAPI contendo o respectivo endereço de comunicação IP e o enumerador.

- **removeProvider**$(CloudProvider)$**:** Após receber um conjunto de informações a respeito dos provedores de nuvens públicas, pode-se remover aqueles que não deseja mais utilizar por meio do enumerador desejado.

- **addHSM**$(hsmIP, hsmIdm, n)$**:** Tem como objetivo adicionar novos MSCs que gerenciarão as chaves criptográficas dos usuários. Para simplificar a arquitetura prevista nesse trabalho, limita-se a utilizar um conjunto

de quatro MSCs, com a intenção de manter o mesmo número de provedores de nuvens públicas. Deve-se especificar o endereço ip do MSC (*hsmIP*), assim como seu identificador (*hsmID*), número máximo e mínimo de MSCs (m,n) no gerenciamento das chaves.

- **returnHSMList:** Retorna um conjunto de identificadores de MSCs que já foram adicionados no middleware.

- **removeHSM**$(hsmId)$**:** Após adicionar um conjunto de endereços para comunicação com diferentes MSCs, pode-se remover aqueles que não deseja-se utilizar passando o identificador do MSC (*hsmId*) obtido na listagem de MSCs.

- **searchForMetadata**$(dataId)$**:** Para o compartilhamento ou recuperação de arquivos sigilosos, deve-se procurar primeiramente pelos metadados de um determinado arquivo para verificar suas informações. Deve-se especificar o nome do arquivo que deseja-se compartilhar. Caso o documento exista nos provedores de nuvens públicas, os metadados serão retornados contendo informações a respeito dos grupos que possuem acesso, assim como as versões existentes no sistema de armazenamento. Caso o documento não exista, é retornado ao usuário um metadado nulo. Para que isso aconteça, um token de autenticação já deve ter sido emitido com o método createHSMUser.

- **sharePrivateData**$(dataPath, Metadata)$**:** Para que um usuário do sistema possa compartilhar um arquivo, deve-se passar o caminho do mesmo (*dataPath*), assim como um metadado ($Metadata$) contendo informações desse arquivo, como nome, versão e grupo que poderá ter acesso ao mesmo. Antes de realizar essa operação, deve-se consultar os metadados relacionados ao arquivo que deseja-se compartilhar para verificar se existem e quais as informações que já estão disponíveis nos provedores de nuvens públicas.

- **recoverPrivateData(**$Metadata$**):** Para recuperar um determinado arquivo, deve-se especificar o metadado ($Metadata$) do arquivo desejado. Assim como no método de compartilhar, deve-se primeiramente tentar obter os metadados do arquivo que se deseja recuperar para verificar quais as especificações do arquivo que se deseja obter.

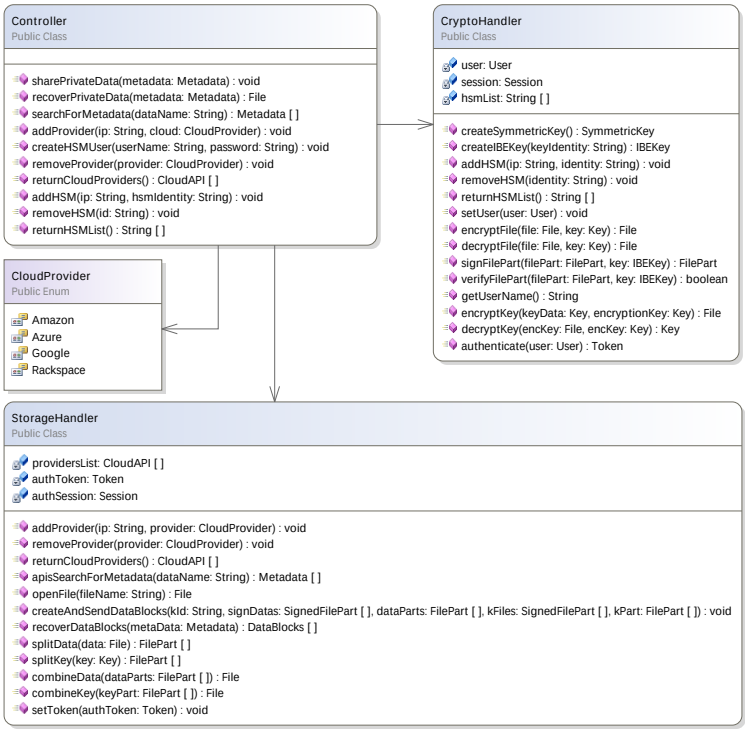

Figura 7 – Diagrama de Classe ilustrando os principais componentes do middleware.

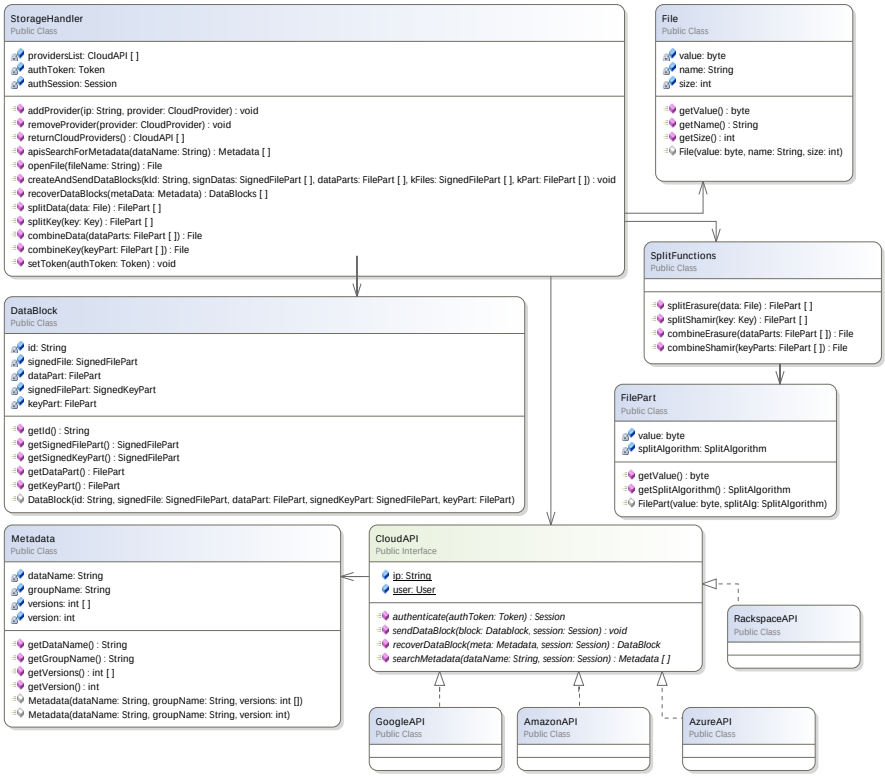

Figura 8 – Diagrama de Classe do Storage Handler.

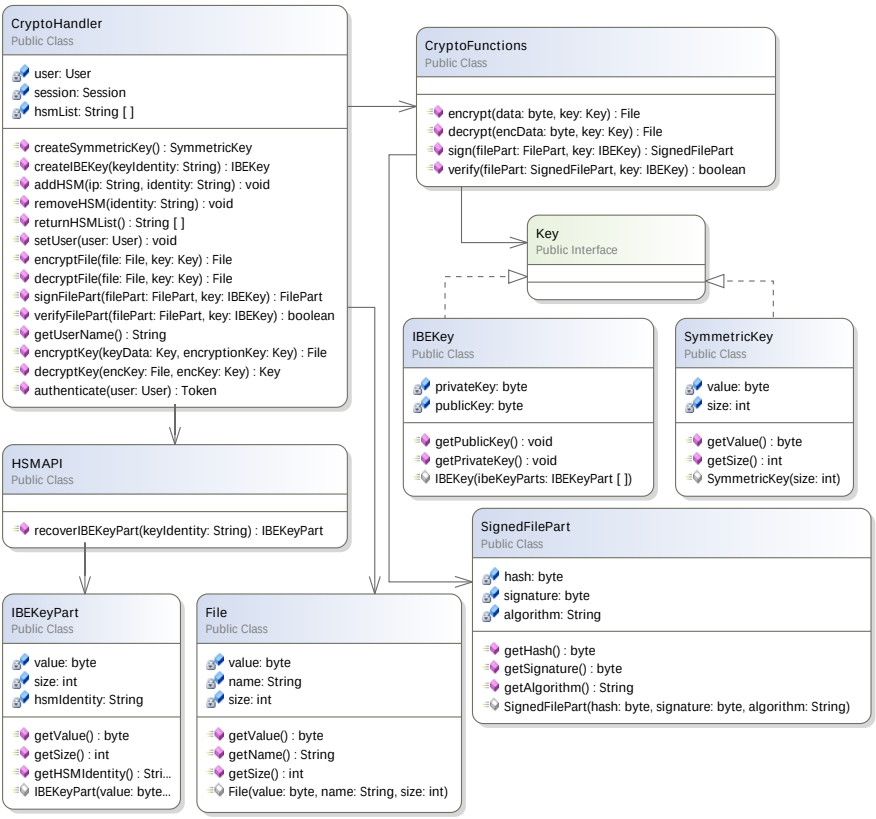

Figura 9 – Diagrama de Classe do Crypto Handler.

5.6.2 Módulo de Segurança Criptográfico

O MSC tem como principal função o correto gerenciamento das chaves criptográficas, protegendo-as de forma física e lógica. Os MSCs devem ser previamente configurados com um firmware contendo os aplicativos necessários para o seu correto funcionamento, incluindo o aplicativo gerenciador de chaves criptográficas. Com esse firmware, o MSC disponibilizará o aplicativo de gerenciamento das chaves privadas baseadas em identidade utilizando APIs próprias para comunicação com os usuários. Essa API deve ser simples o suficiente para não prejudicar a segurança do MSC, mas deve prover todas as funções necessárias para o correto gerenciamento das chaves

criptográficas. As principais chamadas da API que o middleware utilizará são:

- **Authenticate**$(Usuario, Senha)$**:** Realizará a autenticação do usuário previamente cadastrado no sistema de gestão de identidades (SGI). Esse método verificará se existe tal usuário em seu controle de acesso e se a senha fornecida confere com a armazenada anteriormente. Esse procedimento ainda não verificará se o identificador repassado para a criação da chave criptográfica confere com as restrições de acesso do usuário. Após entrar em contato com o SGI e receber as propriedades que o usuário tem acesso, esse método irá entrar em contato também com os provedores de nuvem pública para requisitar os tokens de autenticação. Após essa requisição, os tokens serão enviados para os usuários utilizarem na comunicação com os provedores de nuvem e requisitar metadados, buscar e escrever arquivos.

- **RecoverIBEPart**$(IdentidadeDaChave)$**:** Cada MSC verificará se o identificador repassado pelo usuário autenticado anteriormente confere com suas limitações de acesso. Caso o mesmo possa acessar um determinado documento, pertencente a um determinado grupo, o MSC irá liberar a operação e devolverá uma das partes da chave privada. Caso contrário, não retornará nada que possa comprometer a segurança dos arquivos sigilosos.

Os MSCs devem inicialmente ser configurados para compartilharem partes de uma chave mestra. O aplicativo responsável nos MSCs deverá executar os passos citados na Seção 3.6.3, ou seja, todos os MSCs envolvidos no gerenciamento das chaves criptográficas baseadas em identidade compartilharão uma parte da chave mestra que gerará as chaves privadas dos usuários do middleware. Da mesma forma, os MSCs devem ser configurados para possuirem acesso aos provedores de nuvem com a finalidade de emitir tokens de acesso aos usuários.

Após serem configurados com o firmware específico contendo os aplicativos necessários e realizado o procedimento de compartilhamento das partes da chave mestra, deve-se realizar o procedimento de criação do controle de acesso. Esse procedimento deve ser feito por outra aplicação que terá uma comunicação segura com o MSC, fazendo com que o controle de acesso dos MSCs seja diferente dos provedores de nuvem públicos que armazenarão os arquivos cifrados sigilosos. Dessa forma, caso o controle de acesso das nuvens seja comprometido, a obtenção das chaves privadas se manterá íntegro.

5.6.3 Funções Internas do Middleware

Os diagramas de classe tem como principal objetivo ilustrar a arquitetura das classes que devem ser utilizadas para garantir o bom funcionamento do middleware para compartilhamento de dados sigilosos. Entretanto, necessita-se especificar como essas classes comunicam-se entre si para realizar as funções desejadas. A seguir encontram-se algumas funções internas com seus respectivos diagramas de sequência contendo as interações entre as classes do middleware.

- **Criar Usuários e Buscar Metadados:** Essa é a primeira etapa antes de realizar as próximas funções do middleware. Deve-se primeiramente executar a função que tem como principal objetivo atribuir os usuários e senhas obtidos da aplicação do usuário na classe de gerenciamento de criptografia. Após realizar essa atribuição, os MSCs autenticam o usuário utilizando o sistema de gestão de identidades. Após a correta autenticação, os MSCs entram em contato com os provedores de nuvem para buscar por tokens de autenticação com limitação de tempo e privilégios de acesso. Esse token é então retornado para que o usuário possa consultar por metadados disponíveis dos provedores de nuvens públicas. A Figura 10 ilustra a interação entre as classes para a criação de usuários e a busca de metadados.

- **Criar Chaves:** Ao compartilhar ou recuperar dados sigilosos, deve-se criar chaves criptográficas tanto para cifrar quanto assinar dados de conteúdo sensível. O procedimento de criação de chave simétrica é simples, baseando-se apenas em enviar um tamanho para a sua criação. Para a criação das chaves assimétricas baseadas em identidade, necessita-se de comunicação com os MSCs em nuvens privadas. Para isso, existem uma classe com a API necessária para realizar a comunicação com os MSCs. Devido ao fato de que a mesma aplicação deve estar embarcada em cada MSC, não existe a necessidade de se criar novas APIs de comunicação. A Figura 11 ilustra a interação entre as classes para a criação de chaves criptográficas utilizadas no middleware.

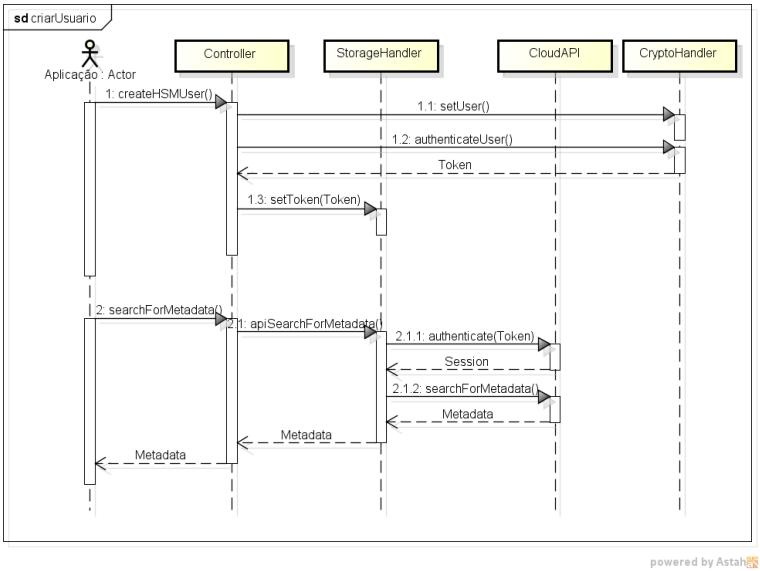

Figura 10 – Diagrama de sequência para a operação de criar usuário interno ao middleware.

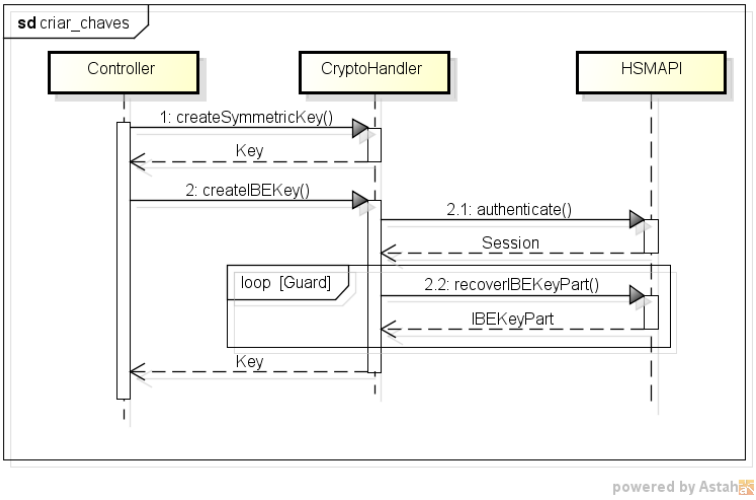

Figura 11 – Diagrama de sequência para a operação de criar chaves.

- **Cifrar e Decifrar dados :** Para garantir a confidencialidade de dados compartilhados utilizando armazenamento em nuvens públicas, deve-se cifrar os dados localmente utilizando-se criptografia simétrica. Dessa forma, garante-se que os provedores de nuvem não tem acesso aos dados sigilosos. Para ter acesso aos dados em claro dos documentos sigilosos cifrados, deve-se decifrá-los localmente com as chaves simétricas correspondentes. A Figura 12 ilustra a interação entre as classes para as operações de ciframento e deciframento.

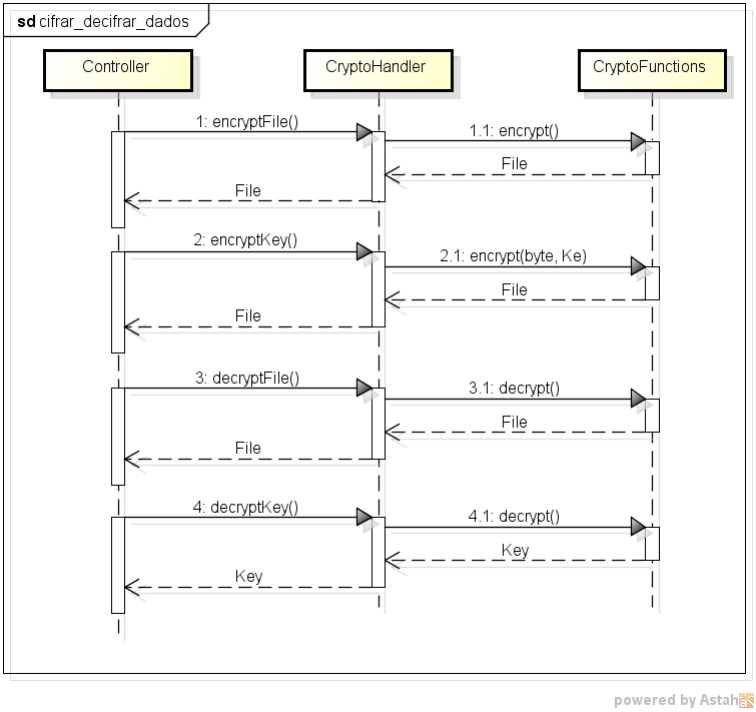

Figura 12 – Diagrama de sequência para as operações de cifrar e decifrar dados.

- **Assinar e Verificar dados :** Para garantir a integridade das partes dos dados sigilosos e chaves armazenadas em provedores de nuvens públicas, deve-se utilizar mecanismos de assinatura digital. Dessa forma, antes de enviar os dados aos provedores de nuvem, os donos dos arquivos devem assinar localmente com suas chaves privadas para garantir o sigilo. No momento da recuperação dos documentos sensíveis, os

usuários devem utilizar as chaves pulicas correspondentes para verificar a integridade das partes dos arquivos. Caso encontre-se um número mínimo de partes íntegras, pode-se recompor o arquivo original com mais segurança de sua integridade. A Figura 13 ilustra a interação entre classes para as operações de assinatura e verificação.

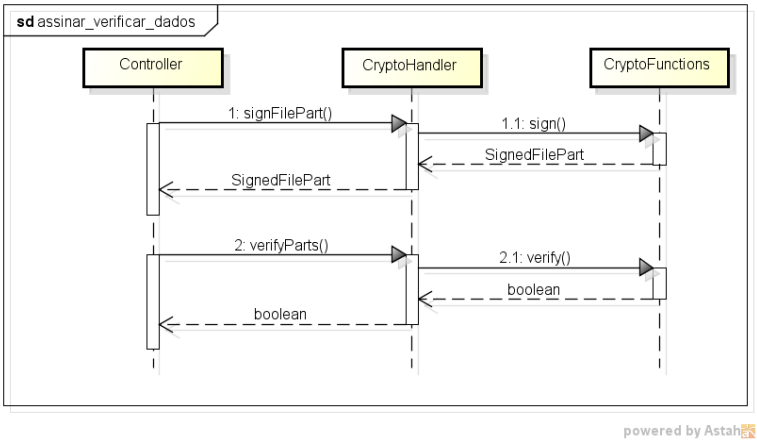

Figura 13 – Diagrama de sequência para as operações de assinar e verificar dados.

- **Quebrar e Combinar partes:** Depois de cifrar os dados sigilosos e as chaves simétricas, deve-se quebrá-los em pedaços para que sejam espalhados nos provedores de nuvens públicas. Para quebrar e recombinar os arquivos cifrados, utiliza-se o *erasure code*, com o intuito de se obter o melhor custo benefício na economia do armazenamento. Para quebrar e recombinar as chaves, utiliza-se o protocolo de segredo compartilhado de Shamir. A Figura 14 ilustra a interação entre as classes para as operações de quebrar e combinar partes.

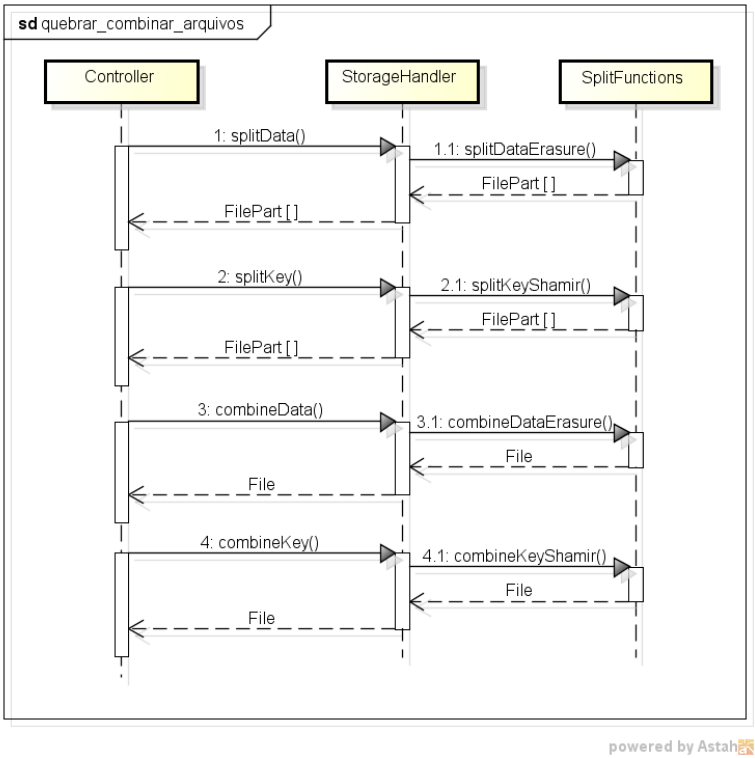

Figura 14 – Diagrama de sequência para as operações de quebrar e combinar dados.

- **Enviar e Receber blocos de dados:** Após os procedimentos em que os dados são cifrados e quebrados, deve-se criar blocos de dados e então enviar para os diferentes provedores de nuvem pública, sendo necessário se autenticar perante os gerenciadores de chaves criptográficas para obter o token de autenticação para cada provedor de nuvem. Cada um dos provedores de nuvem possui uma API diferente que deve ser suportada no middleware previamente. Nesse ponto, os arquivos sigilosos estão armazenados quebrados, cifrados e assinados, juntamente com a chave simétrica que os cifrou. Ao tentar ter acesso aos arquivos sigilosos, deve-se então obter os blocos de dados de diferentes provedores de nuvem, para então extrair os dados de cada bloco afim de continuar o processo de recombinação do arquivo original. A Figura 15 ilustra a interação entre as classes para as operações de criar, enviar

e receber blocos de dados.

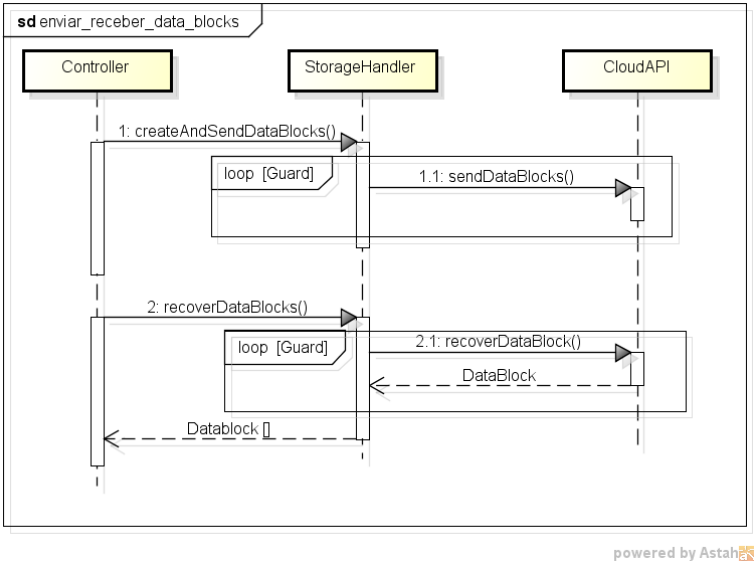

Figura 15 – Diagrama de sequência para as operações de enviar e receber blocos de dados.

- **Compartilhar Dados:** O processo de compartilhar dados consiste nos processos de autenticar perante os gerenciadores de chaves criptográficas, receber o token de autenticação, buscar metadados de um determinado arquivo, criar chaves simétricas e assimétricas, cifrar o documento com a chave simétrica e cifrar a chave simétrica com a chave assimétrica, quebrar os dados com *erasure code* e a chave simétrica com segredo compartilhado de Shamir, assinar as partes do arquivo e da chave simétrica em conjunto com suas assinaturas, criar e enviar os blocos de dados para os diferentes provedores de nuvens públicas. A Figura 16 ilustra a interação simplificada entre as classes para a operação de compartilhar dados.

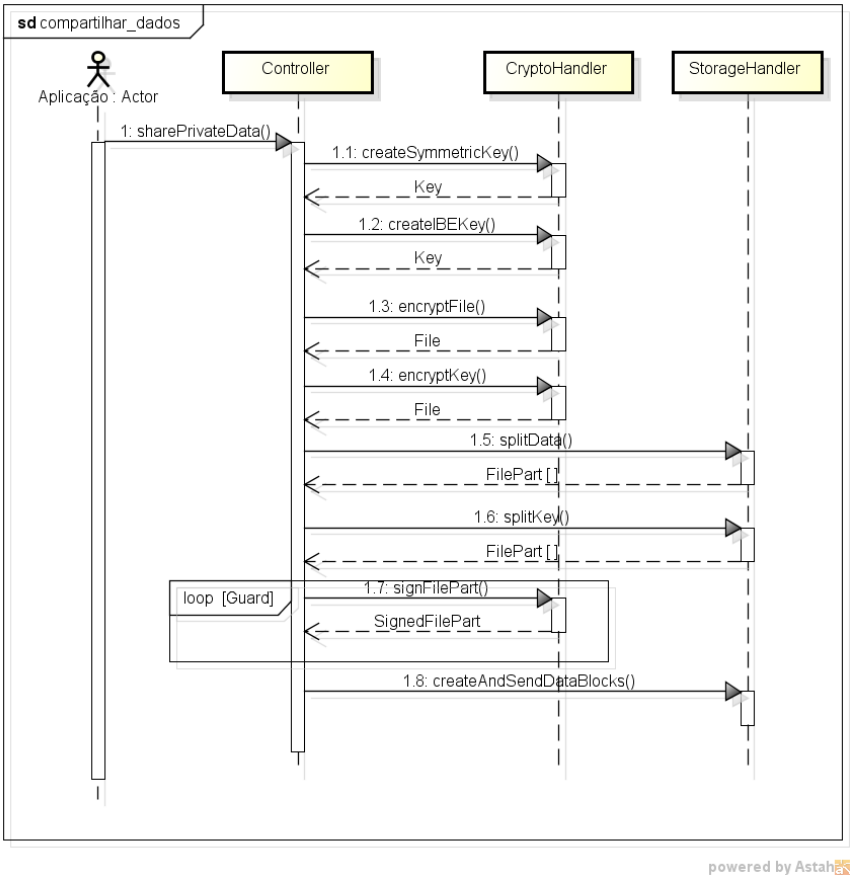

Figura 16 – Diagrama de sequência para a operação de compartilhar dados sigilosos.

- **Recuperar Dados:** O processo de recuperar dados consiste nos processos de autenticar perante os gerenciadores de chaves criptográficas, receber o token de autenticação, receber os blocos de dados, recriar a chave assimétrica, extrair os dados, recombinar a chave simétrica cifrada com segredo compartilhado de Shamir e o arquivo cifrado com o *erasure code* partes que foram quebradas, verificar se os dados cifrados estão íntegros, decifrar a chave simétrica com a chave assimétrica e finalmente decifrar o documento com a chave simétrica. A Figura 17 ilustra a interação entre as classes para a operação de recuperar dados.

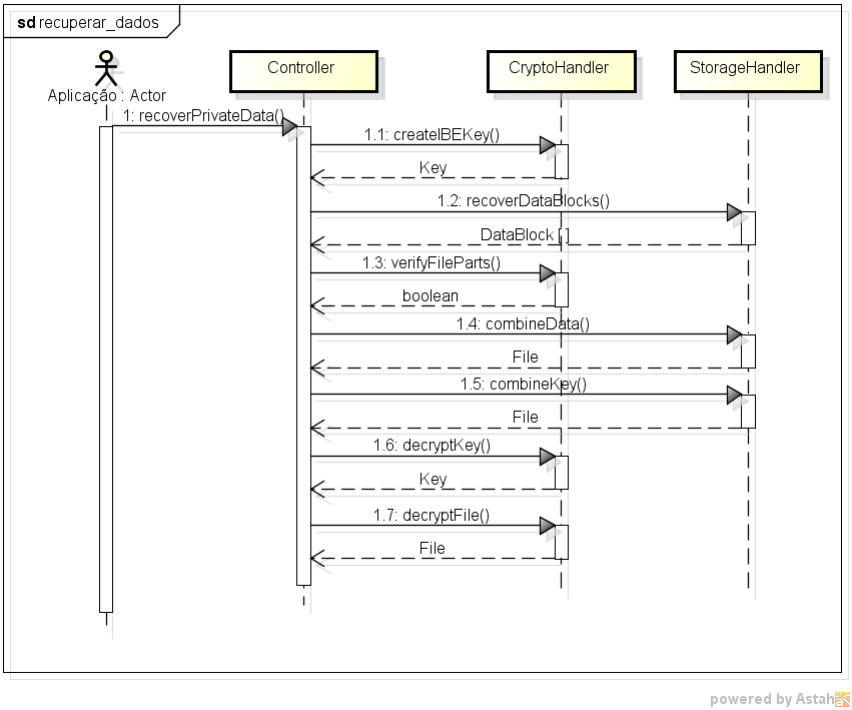

Figura 17 – Diagrama de sequência para a operação de recuperar dados sigilosos.

5.6.4 Algoritmos

Os dois principais algoritmos de uso da arquitetura são o de compartilhar dados na nuvem e outro de recuperar os dados da nuvem. Os dois algoritmos são responsáveis pelos principais processos envolvidos no compartilhamento de documentos sigilosos, como o a criação de chaves criptográficas, ciframento e assinatura de dados e particionamento e recombinação de arquivos.

Este trabalho utiliza o esquema de segredo compartilhado para integrar a confidencialidade com a disponibilidade. Uma vez que todas as chaves simétricas cifradas são quebradas em N pedaços (sendo N o número total de provedores de nuvem), o usuário necessitará de um número mínimo de M partes (M é um número pré-estabelecido no momento da inicialização do sistema) para reconstruir a chave cifrada. De fato, este trabalho reutiliza o

controle de acesso dos provedores de nuvem e dos gerenciadores de chaves criptográficas para controlar quais leitores estarão habilitados para acessar o dado armazenado. Este trabalho também utiliza o mecanismo de *information-optimal erasure code* (PLANK; SIMMERMAN; SCHUMAN, 2008), possibilitando uma economia nos provedores de nuvem ao armazenar diferentes versões do mesmo documento. Do contrário, os custos subiriam por um fator de n se for necessário replicar os arquivos em n provedores de nuvem. Cada parte é reduzida por um fator de $\frac{n}{f+1}$ (RABIN, 1989), considerando f o número de servidores com falhas. Considera-se aqui que o número mínimo de partes para recompor o segredo compartilhado de Shamir está diretamente relacionado com as partes redundantes do *erasure code*. Por exemplo, se considerar um número total de $N = 4$ de nodos e um número mínimo de $M = 2$ de nodos para recuperar uma determinada chave, o *erasure code* irá considerar um total de $T = N - M$ e um número redundante de $R = M$. Sempre será necessário reunir ao menos duas partes de um total para recombinar as partes.

A técnica de Assinatura Hess (HESS, 2003) é utilizada para assinar as partes cifradas das chaves e documentos, e para garantir a integridade. Neste trabalho o mesmo par de chaves é utilizado para cifrar e assinar, facilitando assim o gerenciamento e compartilhamento das chaves. Para os processos que envolvem cifragem e decifragem, este trabalho utiliza o esquema BF-IBE para cifrar chaves simétricas construindo um identificador *ID* específico contendo as seguintes informações: Nome do Documento, Grupo de Custodiantes e Versão do Documento. Formando o seguinte identificador:

```
ID = {nomeDocumento + grupoArquivo + versaoArquivo}
```

Este identificador específico *ID* é utilizado devido a uma série de características que são necessárias para compartilhar documentos sensíveis. Entre eles estão:

1. O Nome do Documento no *ID* é utilizado para que cada documento cifrado e armazenado na nuvem possua uma chave diferente.

2. O Grupo de Custodiantes é para limitar o controle de acesso ao documento; e como estaremos reutilizando o controle de acesso do sistema, este grupo de custodiantes será utilizado para autenticar e liberar o acesso aos documentos sigilosos. Para cada grupo diferente de custodiantes existirá uma chave diferente.

3. O Número de Versão juntamente com os outros elementos são utilizados para controlar o acesso de diferentes versões dos documentos e, desta forma, garantir a segurança na entrada e saída de membros do grupo.

Antes que o usuário possa utilizar o sistema, deve ser realizada a inicialização dos Distribuidores de Chaves Privadas (DCP). O protocolo descrito na seção 3.6.3 é responsável pela geração distribuída da chave mestra da CBI. Após essa etapa, os usuários podem compartilhar documentos sigilosos por meio dos seguintes métodos: *Compartilhar Dado* e *Recuperar Dado*. O primeiro é de responsabilidade do dono do arquivo que vai cifrar, codificar e enviar todas as partes cifradas e assinadas para os provedores de nuvem. O algoritmo *Recuperar Dado* é executado pelos receptores de dados sigilosos que desejam visualizar o conteúdo do documento.

O algoritmo 1 (*Compartilhar Dado*) autentica o usuário, recebe um token de autenticação (linha 4) e solicita para o controle de acesso metadados do documento (linha 5). O novo documento a ser armazenado terá a última versão encontrada (linha 6) mais um (linha 7). Uma chave simétrica é aleatoriamente gerada (linha 8) para cifrar o documento (linha 10). Um identificador *ID* para o documento é definido (linha 11) e uma chave pública baseada neste *ID* é criada (linha 12) utilizando a esquema de BF-IBE. A chave simétrica é cifrada com a chave pública (linha 13) e quebrada em pedaços utilizado o segredo compartilhado de Shamir (linha 14). O documento cifrado é codificado utilizando-se o algoritmo de *information-optimal erasure code* (linha 15), reduzindo o tamanho dos dados que serão armazenados nos provedores de nuvem. A chave privada é criada baseada no identificador *ID* (linha 17) conforme o procedimento descrito na Seção 3.6.4. Para cada parte das chaves e documentos cifrados, serão fornecidos os resumos criptográficos (linhas 19 e 20) e estes serão assinados (linhas 21 e 22) utilizando-se o esquema de assinatura Hess. Um bloco de dados é criado, reunindo toda a informação necessária para armazenar o documento (linha 23). O bloco de dados é enviado para os provedores de nuvem (linha 24) e a entrada deste armazenamento é enviado para o controle de acesso (linha 27). A Figura 18 ilustra de forma simplificada o algoritmo Compartilhar Dado seguindo a numeração de sequência 1 à 6.

Algoritmo 1 COMPARTILHARDADO($nomeArquivo, grupoArquivo, usuario, senha$)

```
1:  total ⟵ n
2:  redundante ⟵ m
3:  dado_ver ⟵ 0
4:  token ⟵ autenticar(usuario, senha)
5:  mt ⟵ buscarMetadados(nomeArquivo, grupoArquivo, usuario, token)
6:  dado_ver ⟵ max(mt[i].ver : 0 ≤ i ≤ n − 1)
7:  dado_ver_novo ⟵ dado_ver + 1
8:  ks ⟵ gerarChaveSim()
9:  dado ⟵ abrirArquivo(nomeArquivo)
10: e_dado ⟵ cifrar(dado, ks)
11: id ⟵ nomeArquivo + "/" + grupoArquivo + "/" + dado_ver_novo
12: pubk_id ⟵ gerar_chave_pub(id)
13: e_ks ⟵ cifrar(ks, pubk_id)
14: enc_ks[0 .. n-1] ⟵ partir(e_ks, total − redundante, total)
15: enc_dado[0 .. n-1] ⟵ codificar(e_dado, total − redundante, redundante)
16: i ⟵ 0
17: privk_id ⟵ gerar_chave_priv(id)
18: for (0 ≤ i ≤ total − 1) do
19:     dado_hash ⟵ H(enc_data[i])
20:     ks_hash ⟵ H(enc_ks[i])
21:     dado_hash_assinado ⟵ assinar(dado_hash, privk_id)
22:     ks_hash_assinado ⟵ assinar(ks_hash, privk_id)
23:     blocoDeDado ⟵ (id, enc_ks[i], enc_dado[i], dado_hash_assinado, ks_hash_assinado)
24:     ack ⟵ mensagemDeEnviarDados(cloud_i, blocoDeDado, token)
25:     if (ack =' ok') then
26:         controleBlocoDeDado ⟵ (id, usuario, grupoArquivo)
27:         enviarMsgControleDeAcesso(cloud_i, controleBlocoDeDado)
28:     end if
29:     i ⟵ i + 1
30: end for
```

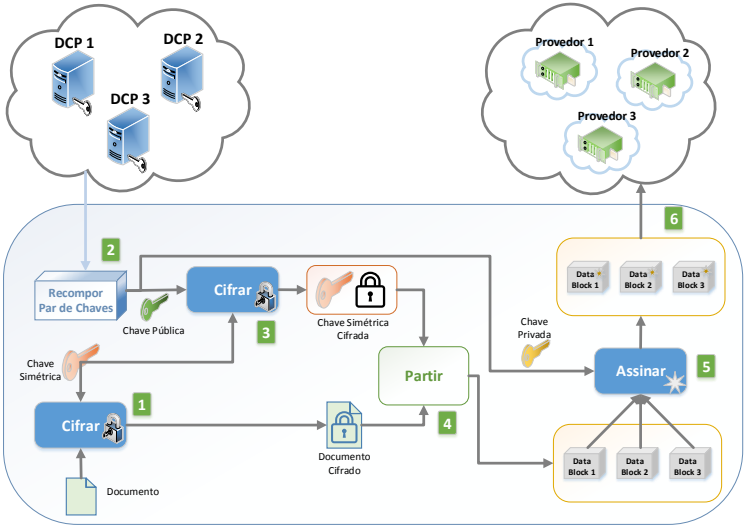

Figura 18 – Ilustração simplificada do algoritmo Compartilhar Dado.

O algoritmo 2 (Recuperar Dado) primeiramente autentica o usuário, recebe o token de autenticação (linha 3) e busca no controle de acesso pelos metadados do documento (linha 4). A última versão é escolhida (linha 5) e o identificador *ID* é composto (linha 6). Um par de chaves é gerado a partir do *ID* utilizando o esquema BF-IBE (linhas 7 e 8) e a requisição pelos dados é iniciada (linha 11). Os blocos de dados são requeridos dos provedores de nuvem (linha 12), no qual cada bloco de dado é verificado a partir das suas assinaturas e resumos criptográficos utilizando o esquema de assinatura Hess (linha 15). Após pegar o número mínimo de blocos de dados (linha 23), as partes do documento cifrado são decodificadas (linha 31), a chave simétrica é recomposta utilizando o segredo compartilhado de Shamir (linha 32), decifrada (linha 33) e finalmente, o documento original é decifrado (linha 34). A Figura 19 ilustra de forma simplificada o algoritmo Recuperar Dado seguindo a numeração de sequência 1 à 6.

Algoritmo 2 RECUPERARDADO($nomeArquivo, grupoArquivo, usuario, senha$)

```
1:  total ⟵ n
2:  redundante ⟵ m
3:  token ⟵ autenticar(usuario, senha)
4:  mt ⟵ buscarMetadados(nomeArquivo, grupoArquivo, usuario, token)
5:  dado_ver ⟵ max(mt[i].ver : 0 ≤ i ≤ n − 1)
6:  id ⟵ nomeArquivo + "/" + grupoArquivo + "/" + dado_ver
7:  privk_id ⟵ gerar_chave_privada(id)
8:  pubk_id ⟵ gerar_chave_pub(id)
9:  i ⟵ 0
10: ERRO ⟵ 0
11: while (i ≤ n − 1) do
12:     t_b ⟵ cloud_i.buscarBlocoDeDado(id, token)
13:     t_eks ⟵ t_b.retorna_enc_ks()
14:     t_edado ⟵ t_b.retorna_enc_data()
15:     rt ⟵ verifica(t_b.ks_hash_assinado_i, t_b.data_hash_assinado_i, t_eks, t_edado, pubk_id)
16:     if (rt = true) then
17:         enc_ks[i] ⟵ t_eks
18:         enc_dado[i] ⟵ t_edado
19:     else
20:         ERRO ⟵ ERRO + 1
21:     end if
22:     i ⟵ i + 1
23:     if (i > redundante − 1) then
24:         Break
25:     else
26:         if (ERRO > redundante − 1) then
27:             retorna ERRO
28:         end if
29:     end if
30: end while
31: e_dado ⟵ decodificar(enc_dado, total − redundante, redundante)
32: e_ks ⟵ combinar(enc_ks, total − dedundante, total)
33: ks ⟵ decifrar(e_ks, privk_id)
34: retorna. Decifrar(e_dado, ks)
```

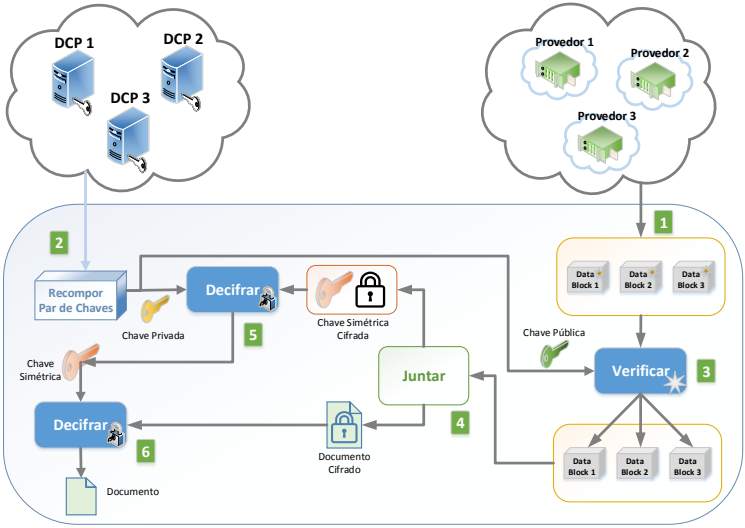

Figura 19 – Ilustração simplificada do algoritmo Recuperar Dado.

5.6.4.1 Implementação dos Algoritmos

A implementação foi dividida em duas partes: A inicialização dos DCPs e os algoritmos de compartilhar e recuperar dados. A implementação foi feita em módulos e serve mais como prova de conceito para validar as ideias aqui propostas do uso da criptografia baseada em identidade em conjunto com outros protocolos criptográficos. Sendo assim, podem ser otimizadas para encontrar melhores resultados.

A inicialização dos DCPs foi implementada por Aniket em seu trabalho (KATE; HUANG; GOLDBERG, 2012) utilizando C++ e o protocolo modificado de JF-DKG. O trabalho de Aniket utiliza a biblioteca de emparelhamento bilinear PBC (*Pairing Based Cryptography*) (LYNN, 2012). As comunicações necessárias entre os clientes e servidores foram implementadas utilizando *sockets* e para a comunicação segura foi utilizada a biblioteca *OpenSSL* (YOUNG; HUDSON; ENGELSCHALL, 2001).

Os algoritmos de Leitura e Escrita foram implementados em C/C++ utilizando as seguintes bibliotecas: *pbc library* (LYNN, 2012) para realizar as operações de emparelhamento bilinear, a biblioteca *gfshare* para realizar as operações de segredo compartilhado de Shamir (MCVITTIE, 2013), a biblioteca *jerasure* para codificar e decodificar utilizando o *information-optimal*

erasure code (PLANK; SIMMERMAN; SCHUMAN, 2008) e a biblioteca *OpenSSL* para realizar algumas das operações usuais de criptografia.

Para a criptografia simétrica foi utilizado o algoritmo AES para cifrar os arquivos com um tamanho de chave de 128 bits. Para o resumo criptográfico foi utilizado o algoritmo SHA-1. Para a cifragem das chaves simétricas foi utilizada a técnica BF-IBE e para assinar as partes cifradas foi utilizada a técnica de assinatura de Hess. A inicialização dos distribuidores de chaves privadas seguiu o parâmetro $3f + 1$, sendo f o número de servidores que podem falhar durante esse procedimento. A implementação utilizou como parâmetros para os algoritmos de segredo compartilhado e erasure code os valores $n = 4$ e $t = 2$, sendo n o número total de partes e t o número mínimo para recompor as partes. A implementação foi dividida em duas etapas: Inicialização dos DCPs/Distribuição da Chave e Execução de Algoritmos.

- **Inicialização dos DCPs/Distribuição da Chave:** O processo de inicialização dos DCPs foi realizado em diferentes máquinas virtuais localizadas no provedor de nuvem da Amazon EC2, gerando assim o segredo compartilhado para a emissão das chaves privadas dos usuários. Utilizou-se a implementação disponível de Aniket para obter as partes da chave privada de um usuário específico fazendo comunicação com as diferentes réplicas do provedor de nuvem da Amazon EC2.

- **Execução de Algoritmos:** A segunda etapa foi realizada localmente, onde a chave privada foi gerada com as partes obtidas anteriormente e foram executados os algoritmos expostos na seção 5.6.4.

A implementação realizada neste trabalho é caracterizada como prova de conceito, não sendo realizado a comunicação na execução dos algoritmos da seção 5.6.4 com os provedores de nuvem. Os algoritmos executam os procedimentos criptográficos e armazenam os documentos localmente para conferência da corretude de execução de cifragem e assinatura e avaliação de desempenho. O código da implementação [1] está disponível para conferência.

5.7 CONCLUSÃO DO CAPÍTULO

Este capítulo apresentou requisitos para a criação de uma arquitetura de middleware para prover o compartilhamento seguro de documentos sigilosos utilizando provedores de nuvens pública para armazenamento e MSCs para o gerenciamento de chaves criptográficas. A partir desses requisitos,

[1] Implementação do protótipo: `http://www.inf.ufsc.br/~rick.lopes/codigo.tar.gz`

pode-se definir premissas que o modelo deve cumprir para o bom funcionamento e para as garantias de seguranças.

A partir dos requisitos e premissas estabelecidas, um modelo foi proposto para o compartilhamento de documentos sigilosos. Esse modelo teve como base o uso da criptografia baseada em identidade distribuído para o gerenciamento de chaves criptográficas e o uso da computação em nuvem para o armazenamento dos dados. O modelo é composto por quatro tipos diferentes de componentes: o usuário final, o provedores de nuvem pública para armazenamento, gerenciadores de chaves criptográficas distribuídos e um sistema de gestão de identidades.

A partir do modelo, pode-se construir a arquitetura do middleware, listando sua API com os principais comandos de acesso para aplicações que desejam prover compartilhamento seguro de documentos sigilosos utilizando MSCs e provedores de nuvem pública para armazenamento. A partir da arquitetura, pode-se construir os diagramas de classe do middleware, em conjunto com seus diagramas de sequência para ilustrar o funcionamento de algumas funcionalidades da arquitetura.

Com o modelo e arquitetura para o compartilhamento de dados sigilosos em nuvem, pode-se criar os algoritmos para o envio e recebimento de dados. Esses algoritmos são os passos necessários de alto nível para funcionamento da arquitetura. Os algoritmos foram listados e explicados em detalhes, assim como figuras ilustrando de maneira simplificada foram listadas para um melhor entendimento geral dos mesmos.

Tendo os algoritmos prontos, pode-se implementá-los utilizando as bibliotecas necessárias em C e C++ para uma futura avaliação dos métodos. A implementação tem a intenção de validar os processos de criptografia baseada em identidade, assinatura e quebra de arquivos. Com esses algoritmos implementados, pode-se realizar avaliações de desempenho do código, que serão apresentadas no capítulo seguinte.

6 AVALIAÇÃO

6.1 INTRODUÇÃO

Este capítulo tem como objetivo realizar avaliações de segurança do modelo proposto e avaliações de desempenho dos algoritmos e implementações realizadas. As avaliações de segurança tem como objetivo investigar os requisitos apresentados no decorrer das pesquisas e verificar que o modelo geral e a arquitetura cumpra com esses requisitos. As avaliações de desempenho tem como principal objetivo verificar a viabilidade dos protocolos propostos para utilização em aplicações reais.

6.2 AVALIAÇÃO DE SEGURANÇA

Primeiramente avaliou-se os aspectos gerais de segurança do modelo que são providos independentemente do caso de uso. O principal aspecto levado em conta é o gerenciamento das chaves criptográficas, que são divididos em:

- Custódia das Chaves
- Revogação Segura das Chaves
- Tolerância a Falhas
- Integridade das Partes
- Arquitetura

6.2.1 Custódia das Chaves

Um dos principais problemas apontado pelo documento SP800-144 (JANSEN; GRANCE, 2011) é a vulnerabilidade de ataques internos e a falta de suporte legal em casos de intrusão devido a localização geográfica dos servidores. Para resolver esse problema, este trabalho propõe o uso de gerenciadores de chaves distribuídos utilizando o protocolo modificado de JF-DKG para gerar a chave mestra sem que nenhuma das autoridades tenha controle total da mesma.

Utilizando gerenciadores de chaves criptográficas distribuídos, dois parâmetros são atribuídos: t e N, tal que N é o número total de autoridades e

t representa o número mínimo de partes que serão necessárias para recuperar a chave privada. Deste modo, um agente malicioso precisa descobrir um total de t partes para recompor o segredo, reduzindo assim as chances de um ataque.

Cada módulo de segurança criptográfico terá posse de apenas uma parte do segredo mestre s_i e com isso, mesmo que obtenham o identificador que será utilizado, terão acesso apenas a uma parte $S_iH(ID)$ do segredo do usuário. Para a reconstrução total da chave, o atacante deverá possuir um número mínimo t. Sem este número mínimo, o atacante não conseguirá executar a equação $D_{id} = \sum_{P_i \in O} \lambda_i s_i H(ID)$ para obter a chave privada do usuário.

Adotando a característica de gerenciadores de chaves criptográficas distribuídos, consegue-se retirar a custódia das chaves privadas dos distribuidores de chaves privadas, problema esse inerente ao protocolo da criptografia baseada em identidade.

6.2.2 Revogação Segura das Chave

Para manter a confidencialidade da informação e evitar problemas com a revogação das chaves privadas, é recomendado utilizar parâmetros adicionais para identificar a chave pública da Criptografia Baseada em Identidade (CBI). Uma parte da solução é não vincular uma chave por usuário, mas vincular grupos de usuários com documentos, tendo assim chaves semânticas.

Este trabalho propõe o uso de identificadores contendo regras de acesso concatenado com o nome do documento e a versão do mesmo. Essas regras são verificadas pelos DCPs por meio do controle de acesso que deve ser realizado de maneira distribuída e de uma forma confiável, ou seja, cada DCP faz a sua verificação conforme o controle de acesso que possui. O nome do documento vincula uma chave pública a um documento específico. A versão faz com que a cada modificação do documento, seja gerado um novo par de chaves. Dessa forma, o identificador terá o seguinte formato:

```
ID = {nomeDocumento + grupoArquivo + versaoArquivo}
```

Essas propriedades são garantidas pelo uso de um algoritmo de resumo criptográfico na geração dos identificadores e chaves, tal que ID será a composição destas regras e a chave privada será definida por: $D_id = sH(ID)$. Portanto, um membro que faz parte de um grupo e obteve acesso à chave privada para decifrar um documento em uma versão X, não conseguirá obter uma chave consequente para decifrar um documento na versão $X+1$ caso não faça mais parte deste grupo. Se um usuário já obteve a chave privada, este em algum momento teve acesso a um documento em uma versão específica.

Dessa forma, não existe a necessidade de recifrar o conteúdo do documento que já foi exposto. Caso um usuário não tenha ainda obtido a chave privada e já não pertence mais a um determinado grupo, este não mais terá acesso às partes das chaves e documentos cifrados, pois o controle de acesso não permitirá mais o acesso devido ao não cumprimento das regras pré-estabelecidas para aquele documento.

6.2.3 Tolerância a Falhas

Utilizando DCPs distribuídos, segredo compartilhado e *erasure code*, consegue-se garantir tolerância a falhas em dois níveis nesta proposta. A primeira está relacionada ao processo de inicialização dos gerenciadores de chaves criptográficas e recuperação das chaves privadas, ou seja, no gerenciamento de chaves criptográficas conforme as seções 3.6.3 e 3.6.4. O segundo nível está no armazenamento de documentos em diferentes provedores de nuvem. Os processos de tolerância a falhas são melhor descritos a seguir:

- **Gerenciamento de Chaves Criptográficas:** Os processos de gerenciamento das chaves criptográficas possuem tolerância a falhas devido ao uso do protocolo de JF-DKG (GENNARO et al., 1999). Esse protocolo é baseado no uso de quorum secreto compartilhado e verificável, aumentando assim o rigor na verificação das partes distribuídas. O sistema baseia-se no uso de um total de n servidores, fazendo que seja necessário um subconjunto t para reconstruir o segredo. Ou seja, o sistema caracteriza-se por ser (n,t), de um total de n servidores, faz-se necessário ter ao menos t servidores para completar o processo de inicialização dos distribuidores de chaves privadas e para recompor as partes das chaves privadas no lado do usuário final. Um número t ou menor não terá acesso ao segredo final.

- **Armazenamento de Documentos:** O armazenamento de documentos nessa proposta depende da utilização de múltiplos provedores de nuvem. Essa abordagem traz o benefício da garantia a tolerância a falhas de um determinado provedor de nuvem. Para que isso seja possível, determina-se um número total de n nodos, criando-se a necessidade de um número mínimo de m nodos para a recuperação do arquivo original. Dessa forma, o *Erasure Code* irá considerar o número total de partes que o arquivo será partido em $T = n - m$, gerando assim um número redundante de m partes. Consequentemente, o sistema consegue oferecer tolerância a falhas no sistema de armazenamento utilizando n provedores de nuvem para o armazenamento das partes dos arquivos utilizados

no sistema.

6.2.4 Integridade das Partes

Para a garantia da integridade das partes, a fim de evitar ataques como modificação das partes ou inserção de novas partes de arquivos, este artigo utiliza o esquema de Assinatura Baseada em Identidade (ABI) de Hess.

O esquema consiste em reutilizar as chaves utilizadas para cifrar as chaves simétricas e assinar as partes que serão armazenadas nos provedores de nuvem. Dessa forma, como mostrado no Algoritmo 2, o usuário só irá reconstruir o arquivo e a chave simétrica cifrada depois que no mínimo t partes passarem pela verificação, considerando um sistema (n,t) composto por um total de n nodos, sendo necessário um mínimo de t para a recuperação do segredo. Caso contrário, pode-se verificar qual parte está corrompida e identificar qual nuvem suspeita estará compartilhando partes dos arquivos não íntegras. Ao identificar a nuvem que está retornando as partes não íntegras, pode-se retirá-la momentaneamente das nuvens confiáveis utilizadas no sistema até que essa se readeque às necessidades de segurança do sistema. Pode-se retirar nuvens do sistema até que ao menos t nuvens estejam disponíveis para uso.

A segurança da assinatura Hess segue a segurança de um esquema genérico de um modelo de oráculo aleatório, baseado no problema de Diffie Hellman no campo dos emparelhamentos utilizados. Utilizando CBI é possível também obter a segurança de textos cifrados escolhidos (*chosen ciphertext security* - IND-CCA) que é uma noção de padrão para esquemas de criptografia pública.

6.2.5 Arquitetura

A arquitetura do middleware apresentada neste trabalho tem como principal característica o uso de nuvens públicas para armazenamento e módulos de segurança criptográficos (MSC). Em ambientes controlados encontram-se diferentes MSCs distribuídos geograficamente para o gerenciamento das chaves privadas, enquanto que o armazenamento das partes dos documentos sigilosos é feito por diferentes provedores de nuvens públicas.

Soluções que adotam confiança total no controle de acesso dos provedores de nuvens públicas possuem um nível de segurança baixo, devido a falta de conhecimento que se tem sobre os processos e procedimentos internos às empresas responsáveis por tais serviços. Existe também a preocupação da

localização geográfica dos provedores públicos, que na maior parte das vezes, encontram-se em outros países, como os Estados Unidos. Esse fato é muito relevante devido às diferenças políticas de cada país que pode acarretar em investigações e espionagem nesses provedores sem a devida autorização dos fornecedores de informações, como usuários, empresas e entidades governamentais. Portanto, soluções que adotam somente as nuvens como ponto de confiança, possuem grande possibilidade de comprometimento dos dados sigilosos.

Esse trabalho limita-se a utilizar os provedores de nuvem de forma distribuída apenas para o armazenamento das partes cifradas dos dados sigilosos e chaves criptográficas simétricas. Nenhum dos provedores de nuvens públicas de armazenamento, como Amazon S3, Microsoft Azure, Google, Rackspace, entre outras, tem acesso ao conteúdo sensível dos usuários, mesmo que todas as nuvens sejam comprometidas. Isso torna o esquema seguro a qualquer tipo de comprometimento de dados utilizando provedores de nuvem pública. A arquitetura ainda prevê tolerância a falhas devido à distribuição dos arquivos e chaves utilizando mecanismos como *erasure code* e segredo compartilhado de Shamir.

O armazenamento das chaves privadas não é necessário nessa proposta devido ao uso da criptografia baseada em identidade. Consequentemente, tem-se a vantagem de não fazer com que os usuários se preocupem em armazenar as chaves em dispositivos criptográficos locais como smartcards ou tokens. O uso desses dispositivos garante a posse única das chaves, entretanto, pode gerar problemas de interoperabilidade e a perda dos mesmos, comprometendo o acesso aos documentos sigilosos.

Outro aspecto importante é a criação das chaves por demanda. Como a criptografia baseada em identidade gera chaves a partir de um conjunto de caracteres, pode-se gerar a mesma chave privada a qualquer momento. Com isso, tem-se a vantagem de obter as chaves somente no momento que for necessário, não havendo a necessidade de armazenamento prévio para o uso.

O controle de acesso nessa proposta é feito de maneira segura e reutiliza o que as entidades já possuem, evitando ter que recriar novas políticas de acesso. Além disso, não é necessário armazenar qualquer tipo de dado dos usuários nos provedores de nuvem pública devido ao uso de tokens de autenticação. Os MSCs autenticam os usuários e conforme as regras de acesso, devolve um token de autenticação para acessar os dados sigilosos que estão armazenados nos provedores de nuvem pública. Dessa forma, ao simplificar o controle de acesso e reutilizando um sistema de gestão de identidades das entidades, obtem-se mais segurança e usabilidade para o usuário final, assim como para a implantação de tal middleware em sistemas existentes.

A utilização de MSCs faz com que a segurança aumente, devido à sua

distribuição geográfica, que deve se restringir à ambientes do mesmo país, e com rígido controle de acesso físico e lógico. Dessa forma, utiliza-se normalmente salas cofres que possuem diversos níveis de segurança físico para evitar que pessoas não autorizadas tenham acesso ao equipamento. Caso uma entidade maliciosa consiga ter acesso físico ao equipamento, o mesmo apresenta diversos mecanismos de segurança, como mencionado na Seção 3.5 que buscam proteger o conteúdo das chaves criptográficas armazenadas internamente. Além disso, mesmo que um desses MSCs seja comprometido, o fato de serem inicializados de forma distribuída, faria com que o serviço não fosse comprometido, devido ao fato de utilizarem segredo compartilhado para a reconstrução das chaves. A Tabela 2 compara os resultados da presente proposta com alguns dos principais trabalhos relacionados.

Procedimento	Gerenciamento Seguro das Chaves	Revogação de Usuários	Tolerância a Falhas
Este Trabalho	✓	✓	✓
(ATENIESE et al., 2006)	✓	✓	X
(XIONG et al., 2012)	✓	✓	X
(PEARSON; SHEN; MOWBRAY, 2009)	✓	X	X
(BESSANI et al., 2013)	X	✓	✓
(RUJ; NAYAK; STOJMENOVIC, 2011)	✓	X	✓
(ZHOU; VARADHARAJAN; HITCHENS, 2011)	X	✓	X

Tabela 2 – Comparação das propriedades alcançadas entre os principais trabalhos relacionados.

6.3 AVALIAÇÃO DE DESEMPENHO

A avaliação do desempenho foi realizada baseando-se no uso dos algoritmos mencionados na sub-seção 5.6.4. Para a avaliação de desempenho, os testes foram executados localmente em um computador com as seguintes características: Processador Intel i3, 4GB RAM com um sistema operacional Linux Ubuntu.

Os testes foram executados com arquivos variando de 1Kbyte até 524288 Kbytes (512 MBytes), incrementados por um fator multiplicativo de 2. Esses números foram utilizados baseando-se na estimativa do uso primário da proposta, que seria para dados comuns, como documentos de texto em formatos PDF e TXT, assim como arquivos maiores, como vídeos em formato AVI ou imagens de sistemas operacionais no formato ISO.

Conforme pesquisa realizada pela Symantec (EWEEK, 2012), por volta de 93% dos arquivos que são compartilhados nos meios corporativos possuem menos de 1GB. Portanto, a faixa de avaliação está em conformidade com o mais provável uso de grande parte dos usuários finais.

Cada algoritmo foi executado 10 vezes e a média dos resultados foi avaliada conforme o desvio padrão. Entre os tamanhos de 1Kbyte e 65536 bytes (64 Mbytes) o tempo de execução foi instável, apresentando desvio padrão superior a 5%. Isto se deve ao fato de que os arquivos nessa faixa de tamanho possuem uma execução rápida, gerando assim dados com uma variação alta de velocidade. O desempenho entre os arquivos de 1 Kbyte a 64 Mbytes possuem uma variabilidade de até 300 milissegundos. A partir do tamanho de 64 Mbytes os tempos começam a crescer linearmente, dobrando conforme o tamanho do arquivo. A Figura 20 ilustra o comportamento conforme o aumento do tamanho dos arquivos entre 1 Kbyte e 8 Mbytes e a Figura 21 ilustra o desempenho dos arquivos entre 16 Mbytes e 512 Mbytes.

Dependendo da aplicação que utilizar os algoritmos, a grande parte dos arquivos cifrados e enviados para a nuvem não terão problemas de desempenho. Por exemplo, aplicações compartilhando arquivos de texto como PDFs com tamanhos menores que 64 Mbytes apresentarão bom desempenho, com até 4 segundos para a execução dos algoritmos. Nos casos onde existe a necessidade de se compartilhar documentos maiores partindo de arquivos com 64 Mbytes, como imagens de alta resolução e vídeos, o desempenho é satisfatório, podendo atingir o tempo de até 32 segundos para buscar e decifrar os dados e 29 segundos para cifrar e enviar os dados.

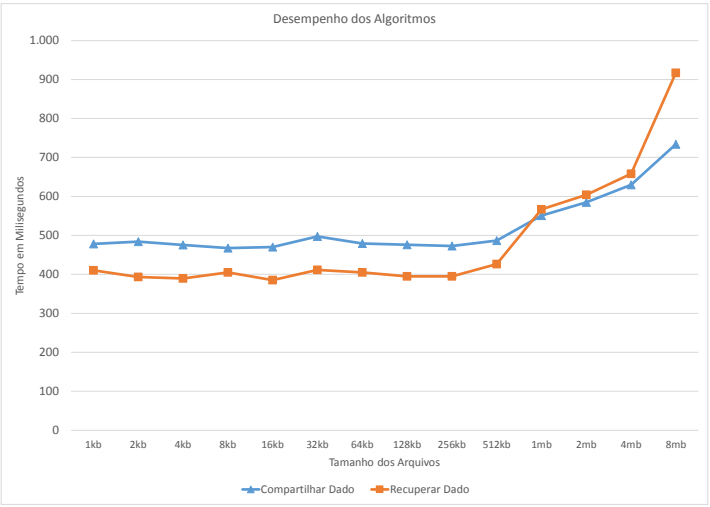

Figura 20 – Desempenho dos algoritmos de Escrever Dado e Ler dado para arquivos entre os tamanhos de 1 Kbyte e 8 Mbytes.

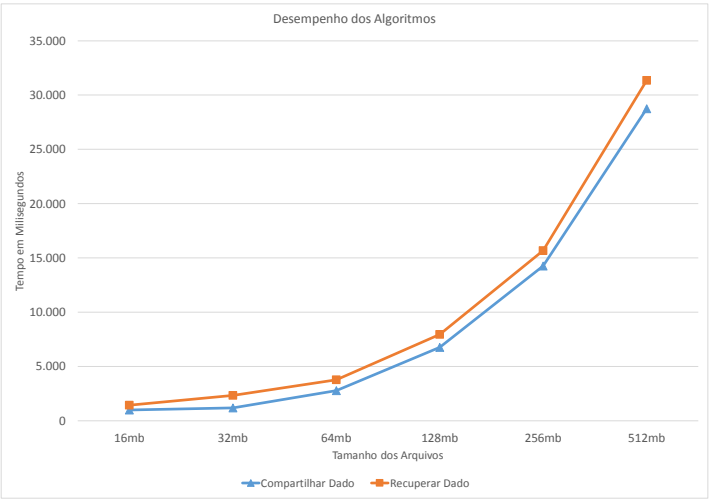

Figura 21 – Desempenho dos algoritmos de Escrever Dado e Ler dado para arquivos entre os tamanhos de 16 Mbytes e 512 Mbytes.

Cada algoritmo possui passos que influenciam no seu desempenho geral, como abrir um arquivo que está armazenado em disco, cifrar o arquivo em memória, obter o resumo criptográfico (*hash*) e assinar, cifrar a chave simétrica, utilizar o *erasure code* para codificar o arquivo cifrado e escrever arquivos cifrados codificados em disco. As Figuras 22 e 23 ilustram as porcentagens de tempo que cada operação principal possui do total da execução dos algoritmos de Escrever e Ler dados.

A partir das figuras mencionadas anteriormente, pode-se concluir que as etapas de leitura e escrita de arquivos em disco consome a maior parte do tempo de execução dos algoritmos. Dessa forma, para uma melhora significativa dos algoritmos, necessita-se de novos trabalhos futuros para analisar o código e algoritmo para uma melhor otimização. Assim como analisar que em um caso real, no algoritmo de Escrita de dados não haverá necessidade de escrever os dados em disco, e sim enviá-los para os provedores de nuvem. Dessa forma, dados como latência e velocidades de Internet influenciam na análise de desempenho dos algoritmos propostos neste trabalho, entretanto, estão fora do escopo deste trabalho.

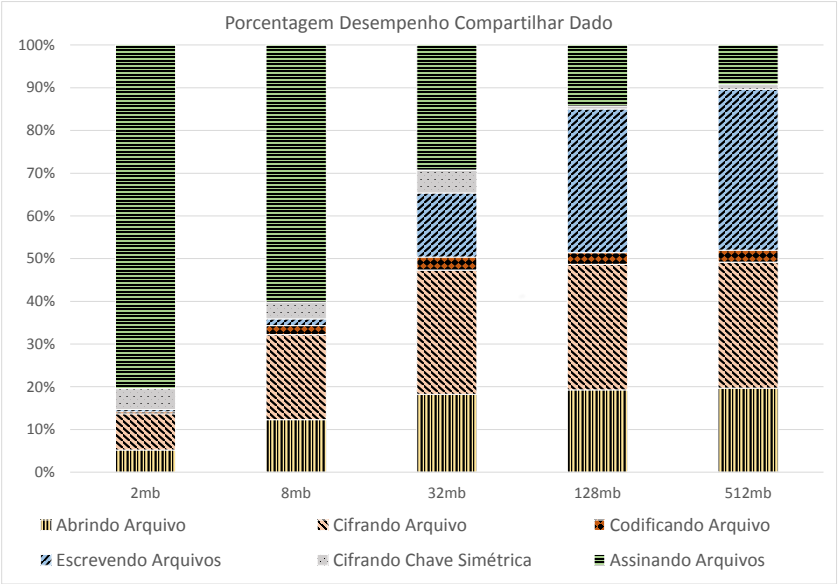

Figura 22 – Porcentagem do tempo gasto para as principais etapas do algoritmo de Escrever Dados.

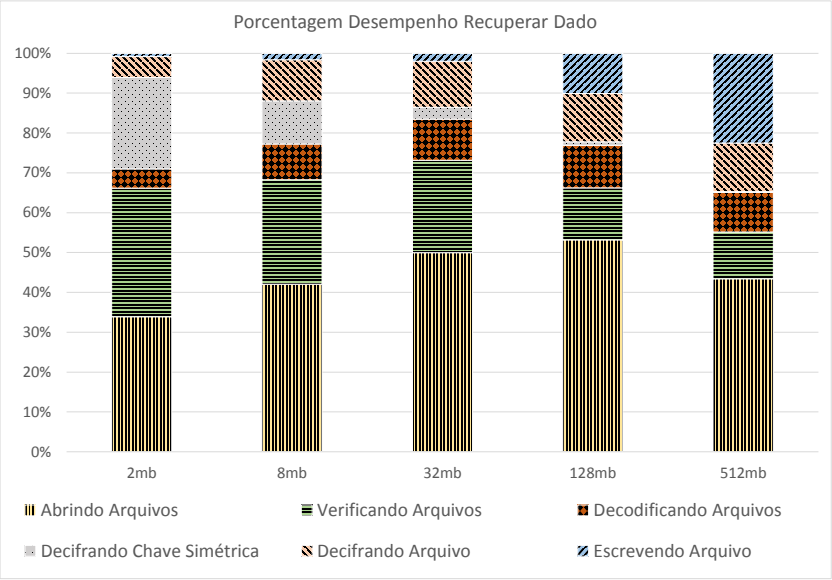

Figura 23 – Porcentagem do tempo gasto para as principais etapas do algoritmo de Ler Dados.

6.4 CONCLUSÃO DO CAPÍTULO

Este capítulo apresentou avaliações de segurança e desempenho dos modelos propostos no capítulo anterior. A partir das avaliações pode-se concluir que os objetivos foram alcançados pela arquitetura proposta, como as propriedades de segurança garantidas e a implementação dos algoritmos.

Por meio da avaliação de segurança pode-se verificar que o mesmo atende aos requisitos impostos e apresentados na Seção 5.2. O compartilhamento de dados sigilosos foi garantido por meio do uso de armazenamento em nuvens públicas, facilitando assim o acesso por terceiros em qualquer dispositivo, necessitando apenas de acesso a internet. A garantia do sigilo foi assegurada por meio do correto e seguro gerenciamento das chaves de criptografia baseada em identidade.

A criptografia baseada em identidade foi amplamente utilizada para um gerenciamento de chaves criptográficas mais seguro e eficiente, proporcionando um ambiente com mais usabilidade para usuários finais. O processo de inicialização distribuída da chave mestre dos distribuidores de chaves pri-

vadas de criptografia baseada em identidade faz com que o benefício e a segurança de nenhuma entidade ter o controle total sobre sua chave privada seja obtido.

Outro objetivo alcançado foi a garantia de segurança no caso de *forward secrecy*, no qual um usuário não deve ter mais acesso aos documentos após a sua saída do grupo. Essa propriedade foi garantida com o gerenciamento seguro das chaves por meio do controle de acesso e distribuição das chaves criptográficas. O objetivo de garantir a segurança nos casos de *backward secrecy*, em que um usuário não pode ter acesso a um documento antes de fazer parte de um grupo, também é garantido por meio do gerenciamento seguro das chaves criptográficas e garantia de segurança em casos de conluio.

A utilização de módulos de segurança criptográficos aumenta a segurança da arquitetura, pois não necessita-se confiar inteiramente nos provedores de nuvem públicas para armazenamento. A partir da avaliação da arquitetura proposta neste trabalho, pode-se concluir que a mesma apresenta um bom nível de segurança contra ataques físicos e lógicos, assim como garante-se a segurança no caso de espionagem e fornecimento de dados a terceiros sem a devida autorização.

Evidenciou-se uma escassez em grande parte dos trabalhos da literatura por implementações reais dos protocolos e avaliações de desempenho. Esse trabalho deixou disponível o código fonte utilizado nos testes e que teve como principal foco uma prova de conceito. Evidenciou-se que a implementação é eficiente para arquivos com até 64 Mbytes de tamanho, fazendo com que o tempo de execução do algoritmo seja quase imperceptível com tempo de até 3 segundos. Como a implementação não utilizou comunicação com provedores de nuvem, esses tempos podem variar para mais ou menos dependendo da latência e velocidade da internet de cada usuário. Devido à escassez de implementações reais dos resultados dos trabalhos relacionados, não foi possível comparar o desempenho dos algoritmos do presente trabalho com outros.

7 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo propor uma arquitetura de middleware para integrar serviços de gerenciamento de chaves criptográficas para sigilo e compartilhamento de dados utilizando provedores de nuvens públicas. A partir do estudo feito, pode-se listar os requisitos necessários para a elaboração de uma arquitetura para o middleware utilizando módulos de segurança criptográficos e provedores de nuvens públicas para armazenamento. Como visto, para os requisitos encontrados, a arquitetura consegue atender as principais necessidades de segurança envolvidos no processo de gerenciamento das chaves e documentos. Protocolos para a realização do gerenciamento seguro das chaves criptográficas e compartilhamento de dados sigilosos foram propostos para cumprir com os requisitos listados.

No Capítulo 3 foram apresentadas fundamentações de segurança para o entendimento das necessidades de um sistema de compartilhamento de dados sigilosos. O estudo e entendimento das ameaças à computação em nuvem determinaram as abordagens e uso de ferramentas de segurança como distribuição dos gerenciadores de chaves criptográficas, uso de módulos de segurança criptográficas, assim como a quebra e distribuição dos arquivos. Problemas e desafios relacionados ao gerenciamento de chaves criptográficas também foram estudados para uma melhor compreensão da problemática envolvida. A partir desses estudos, requisitos foram listados para serem atendidos no modelo do presente trabalho.

O Capítulo 4 apresentou as principais estratégias atualmente abordadas para o compartilhamento de documentos sigilosos em nuvem. O presente trabalho fez um levantamento das principais abordagens dividindo em três tipos diferentes: sigilo como serviço, gerenciadores locais e trabalhos baseados no uso da criptografia baseada em identidade. Como visto, tais trabalhos apresentam limitações quanto ao gerenciamento das chaves criptográficas em conjunto com o uso dos provedores de nuvens públicas. Grande parte desses trabalhos não consegue contornar de maneira conjunta os desafios de *forward secrecy*, *backward secrecy*, tolerância a falhas, tolerância a intrusão e custódia das chaves privadas. Grande parte dos trabalhos relacionados não dispõem de uma análise mais específica do desempenho em casos reais, tendo mais foco nos protocolos matemáticos.

Com os estudos que foram realizados no Capítulo 3 obteve-se requisitos que foram utilizados para analisar os trabalhos relacionados, e com isso, pode-se criar premissas e requisitos para a construção da arquitetura. O Capítulo 5 trata da arquitetura proposta neste trabalho que realiza a integração de serviços de gerenciamento de chaves, funções criptográficas e comparti-

lhamento em provedores de nuvem. Para contornar os desafios encontrados e atender os requisitos estabelecidos, o trabalho abordou primeiramente um modelo geral da arquitetura que foi obtido a partir dos requisitos e premissas definidas. A partir do modelo estabelecido, pode-se mitigar os desafios e especializar em uma arquitetura mais completa para prover os processos necessários para atender os requisitos. A arquitetura final utiliza módulos de segurança criptográficos para o gerenciamento seguro das chaves, fornecendo proteções físicas e lógicas, e provedores de nuvens públicas para o armazenamento das partes dos arquivos sigilosos. A partir da arquitetura, uma implementação de prova de conceito foi feita para avaliar o desempenho dos algoritmos utilizados no middlware.

Após a elaboração da arquitetura, esta foi avaliada no Capítulo 6 de duas formas: Segurança e Desempenho. A avaliação de segurança baseou-se em uma forma argumentativa, demonstrando como cada desafio foi superado com as características da arquitetura. Para comprovar as avaliações, análises teóricas foram realizadas, cujos resultados foram demonstrados com a implementação básica da arquitetura.

Com a arquitetura proposta atingiu-se os objetivos propostos inicialmente de criar um middleware cujas funcionalidades integrassem os serviços de gerenciamento seguro de chaves criptográficas juntamente com o compartilhamento de documentos em provedores de nuvem pública. Para o gerenciamento das chaves utilizou-se módulos de segurança criptográficos distribuídos, fazendo com que os mesmos possuam segurança física e lógica, garantindo a integridade das chaves mestras utilizadas para gerar as chaves privadas dos usuários. Para a segurança e tolerância a falhas no armazenamento das partes dos documentos, a arquitetura utiliza diferentes provedores de nuvem pública para armazenamento. Dessa forma, ao utilizar provedores de nuvem pública para armazenamento em conjunto com módulos de segurança criptográficos, consegue-se garantir o nível de segurança necessário para o compartilhamento seguro de documentos sigilosos.

A garantia de segurança em caso de saída e entrada de membros dos grupos foi garantida por meio do gerenciamento de grupos juntamente com o nome utilizado na geração das chaves criptográficas baseadas em identidade. Ao utilizar um identificador que é composto por nome do documento, grupo custodiante e versão, consegue-se alcançar um nível de unicidade suficiente para a garantia de segurança nos casos de saída e entrada de um novo membro. Dessa forma, o controle de acesso juntamente com as propriedades de segurança da criptografia baseada em identidade garantem a segurança para *forward secrecy* e *backward secrecy*.

O uso de um geradores de chaves privadas distribuídos garante segurança contra a custódia das chaves privadas e também garante tolerância a

falhas. Dessa forma, ao estabelecer um procedimento de inicialização dos geradores de chaves privadas (n,t), consegue-se estabelecer um número mínimo t de partes necessárias para recompor as chaves privadas, tal que $n \geq t$, de tal forma que apenas o usuário que se autenticar de maneira correta em t gerenciadores de chaves possa recuperar sua chave. Isso também garante a tolerância a falhas, devido ao fato de que caso o sistema feito seja $n = 4$ e $t = 3$, mesmo com um servidor comprometido, pode-se conseguir recompor a chave privada utilizando os outros três servidores.

O desenvolvimento deste trabalho resultou nas seguintes publicações em eventos da área:

- Lopes de Souza, R., Vescovi Netto, H., Cheuk Lung, L., & Felipe Custodio, R. (2014, February). SSICC: Sharing Sensitive Information in a Cloud-of-Clouds. In ICONS 2014, The Ninth International Conference on Systems (pp. 185-191). (Qualis B3)

- Nogueira, H., de Souza, R. L., & Custódio, R. F. (2013, October). A Privacy-Enhanced User-Centric Identity and Access Management Based on Notary. In ICSNC 2013, The Eighth International Conference on Systems and Networks Communications (pp. 159-164). (Qualis B3)

- Lopes de Souza, R., Lung, L. C., & Custódio, R. F. (2013, July). Multifactor Authentication in Key Management Systems. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on (pp. 746-752). IEEE. (Equivalente Qualis B2)

7.1 TRABALHOS FUTUROS

Neste trabalho foram abstraídos alguns pontos que podem servir de ponto de partida para trabalhos futuros. Um desses pontos é a implementação completa da arquitetura utilizando provedores de nuvens públicas nas operações de armazenamento. Essa implementação foi abstraída devido ao foco do trabalho ser na segurança do gerenciamento das chaves criptográficas. Uma nova implementação contendo interações com provedores de nuvens públicas pode ser útil para analisar a latência e velocidade encontrada utilizando diferentes provedores. Pode-se observar as diferenças e desafios encontrados para se adaptar aos *storages* de diferentes servidores, diferentes tipos de controle de acesso e diferentes políticas de acesso para cada provedor.

Uma aplicação real, utilizando provedores de nuvens públicas com uma interface amigável seria um bom trabalho futuro para avaliar a usabilidade da aplicação. Pode-se deixar disponível para uso, a fim de realizar

questionários para avaliar a aplicação como um todo, desde a parte da usabilidade e simplicidade da solução, como a questão do desempenho e segurança dos modelos.

De forma mais geral, pode-se realizar provas formais dos protocolos propostos para averiguar a segurança dos mesmos e verificar novas aplicações dos protocolos propostos. Este trabalho menciona duas arquiteturas que podem ser utilizadas em diversos meios e um estudo mais específico de uso agregaria mais qualidade ao estudo feito.

REFERÊNCIAS

ADOBE. It's next challenge: Three key trends in document collaboration and exchange. **[Citado em 22 abril de 2012] Disponível em: http://www.adobe.com/content/dam/Adobe/en/products/acrobat/pdfs/acrobatX_it_challenge.pdf**, 2011.

ANDERSON, R.; MOORE, T. The economics of information security. **Science**, American Association for the Advancement of Science, v. 314, n. 5799, p. 610–613, 2006.

ATENIESE, G. et al. Improved proxy re-encryption schemes with applications to secure distributed storage. **ACM Transactions on Information and System Security (TISSEC)**, ACM, v. 9, n. 1, p. 1–30, 2006.

BARKER, E. et al. Recommendation for key management. **NIST special publication**, v. 800, p. 57, 2011.

BESSANI, A. et al. Depsky: dependable and secure storage in a cloud-of-clouds. **ACM Transactions on Storage (TOS)**, ACM, v. 9, n. 4, p. 12, 2013.

BLAZE, M.; BLEUMER, G.; STRAUSS, M. Divertible protocols and atomic proxy cryptography. In: **Advances in Cryptology-EUROCRYPT'98**. [S.l.]: Springer, 1998. p. 127–144.

BONEH, D.; FRANKLIN, M. Identity-based encryption from the weil pairing. In: SPRINGER. **Advances in Cryptology-CRYPTO 2001**. [S.l.], 2001. p. 213–229.

CHOW, S. S. et al. Dynamic secure cloud storage with provenance. In: **Cryptography and Security: From Theory to Applications**. [S.l.]: Springer, 2012. p. 442–464.

CISCO. Collaboration return on investment. **[Citado em 22 abril de 2012] Disponível em: https://communities.cisco.com/docs/DOC-16566**, 2010.

DENNING, R.; ELIZABETH, D. **Cryptography and data security**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1982.

DIFFIE, W.; HELLMAN, M. New directions in cryptography. **Information Theory, IEEE Transactions on**, IEEE, v. 22, n. 6, p. 644–654, 1976.

ELLISON, C.; SCHNEIER, B. Ten risks of pki: What you're not being told about public key infrastructure. **Comput Secur J**, v. 16, n. 1, p. 1–7, 2000.

EMURA, K. et al. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: **Information Security Practice and Experience**. [S.l.]: Springer, 2009. p. 13–23.

EWEEK. Small businesses at risk from online file sharing: Symantec. **[Citado em 22 abril de 2012] Disponível em: http://www.eweek.com/c/a/Midmarket/Small-Businesses-at-Risk-from-Online-FileSharing-Symantec-624775/**, 2012.

FIPS, P. 140-2: Security requirements for cryptographic modules. **National Institute of Standards and Technology**, 2001.

GENNARO, R. et al. Secure distributed key generation for discrete-log based cryptosystems. In: SPRINGER. **Advances in Cryptology - EUROCRYPT - 99**. [S.l.], 1999. p. 295–310.

GREENBERG, A. Cloud computingś stormy side. **Forbes Magazine**, v. 19, 2008.

GREENWALD, G.; MACASKILL, E. Nsa prism program taps in to user data of apple, google and others. **The Guardian**, v. 7, n. 6, 2013.

HESS, F. Efficient identity based signature schemes based on pairings. In: SPRINGER. **Selected Areas in Cryptography**. [S.l.], 2003. p. 310–324.

ITI, I. N. de Tecnologia da I. Tecnologia verde: menos papel, mais segurança e respeito ao meio ambiente! **[Citado em 22 abril de 2012] Disponível em: http://www.iti.gov.br/images/publicacoes/revista-digital/revista20112.pdf**, 2011.

JAEGER, P. T. et al. Where is the cloud? geography, economics, environment, and jurisdiction in cloud computing. **First Monday**, v. 14, n. 5, 2009.

JANSEN, W.; GRANCE, T. Nist sp 800-144 draft: guidelines on security and privacy in public cloud computing, security division. **Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD**, p. 20899–8930, 2011.

JUNG, T. et al. Privacy preserving cloud data access with multi-authorities. In: **IEEE INFOCOM**. [S.l.: s.n.], 2013.

KAMARA, S.; LAUTER, K. Cryptographic cloud storage. In: **Financial Cryptography and Data Security**. [S.l.]: Springer, 2010. p. 136–149.

KATE, A.; HUANG, Y.; GOLDBERG, I. Distributed key generation in the wild. **IACR Cryptology ePrint Archive**, v. 2012, p. 377, 2012.

KERCKHOFFS, A. **La cryptographie militaire**. [S.l.]: University Microfilms, 1883.

KHAN, D. **The Codebreakers: The History of Secret Writing.[Sl]**. [S.l.]: McMillan Publishing Co, 1967.

LYNN, B. The pairing-based cryptography (pbc) library. **Available on http://crypto. stanford. edu/pbc**, 2012. Disponível em: <http://crypto. stanford. edu/pbc>.

MCVITTIE, S. A secret sharing library - libgfshare. **[Citado em 22 abril de 2012] Disponível em: https://launchpad.net/libgfshare**, 2013.

MELL, P. M.; GRANCE, T. Sp 800-145. **The NIST Definition of Cloud Computing, National Institute of Standards & Technology, Gaithersburg, MD**, 2011.

MESSMER, E. Are security issues delaying adoption of cloud computing. **Network World**, v. 27, 2009.

PEARSON, S.; SHEN, Y.; MOWBRAY, M. A privacy manager for cloud computing. In: **Cloud Computing**. [S.l.]: Springer, 2009. p. 90–106.

PLANK, J. S.; SIMMERMAN, S.; SCHUMAN, C. D. Jerasure: A library in c/c++ facilitating erasure coding for storage applications-version 1.2. **University of Tennessee, Tech. Rep. CS-08-627**, v. 23, 2008.

RABIN, M. O. Efficient dispersal of information for security, load balancing, and fault tolerance. **Journal of the ACM (JACM)**, ACM, v. 36, n. 2, p. 335–348, 1989.

RUJ, S.; NAYAK, A.; STOJMENOVIC, I. Dacc: Distributed access control in clouds. In: IEEE. **Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on Trust**. [S.l.], 2011. p. 91–98.

SCHNEIER, B. The psychology of security. In: **Progress in Cryptology–AFRICACRYPT 2008**. [S.l.]: Springer, 2008. p. 50–79.

SHAMIR, A. Identity-based cryptosystems and signature schemes. In: SPRINGER. **Advances in cryptology**. [S.l.], 1985. p. 47–53.

SHIREY, R. **RFC 4949–Internet Security Glossary**. [S.l.]: Version, 2007.

SMITH, S. W.; WEINGART, S. Building a high-performance, programmable secure coprocessor. **Computer Networks**, Elsevier, v. 31, n. 8, p. 831–860, 1999.

SUTIL, J. M. Gestão segura de múltiplas instâncias de uma mesma chave de assinatura em autoridades certificadoras. **Dissertação (Mestrado em Ciência da Computação) - Departamento de Informática e Estatística, Universidade Federal de Santa Catarina**, p. 119, 2011.

WEEK, I. Time to think about cloud computing. **[Citado em 22 abril de 2012] Disponível em: http://www.informationweek.com/cloud-computing/software/time-to-think-about-cloud-computing/211300562**, 2008.

WHITMAN, M. E.; MATTORD, H. J. **Principles of information security**. [S.l.]: Cengage Learning, 2010.

WILLIAM, S.; STALLINGS, W. **Cryptography and Network Security, 4/E**. [S.l.]: Pearson Education India, 2006.

XIONG, H. et al. Cloudseal: End-to-end content protection in cloud-based storage and delivery services. In: **Security and Privacy in Communication Networks**. [S.l.]: Springer, 2012. p. 491–500.

YAN, L.; RONG, C.; ZHAO, G. Strengthen cloud computing security with federal identity management using hierarchical identity-based cryptography. In: **Cloud Computing**. [S.l.]: Springer, 2009. p. 167–177.

YOUNG, E. A.; HUDSON, T. J.; ENGELSCHALL, R. S. Openssl. **[Citado em 22 abril de 2012] Disponível em: http://www.openssl.org/**, 2001.

ZHOU, L.; VARADHARAJAN, V.; HITCHENS, M. Enforcing role-based access control for secure data storage in the cloud. **The Computer Journal**, Br Computer Soc, v. 54, n. 10, p. 1675–1687, 2011.